République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Frère Mentouri – Constantine 1
Faculté des Sciences de la Nature et de la vie
Département de Biochimie et de Biologie Moléculaire et Cellulaire

Mémoire de Fin d'Etudes Pour l'Obtention du Diplôme de Master
Spécialité : Biochimie Appliqué

Présenté par : BADAOUI Mohamed Yahia Zeriab

Thème

Génération *in-silico* des molécules a visées thérapeutiques basée sur les méthodes d'intelligence computationnelle

Devant le jury :

Président : M. BENSEGUENI. A.	Professeur	U. Frères Mentouri – Constantine 1
Encadreur : M. DEMS.M. A.E.	Maitre de Recherche A	C.R.Bt - Constantine
Co-Encadreur M. BOUKLIA. A.	Maitre de Recherche B	C.R.Bt - Constantine
Examinateur : M. MOKRANI.E.	MAA	U. Frères Mentouri– Constantine 1

Année universitaire : 2019-2020

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon encadreur M.DEMS, Maitre de Recherche A au C.R.B.t – Constantine, et mon Co-encadreur M. BOUKLIA, Dr au C.R.B.t – Constantine ainsi que M.BOUHEDJAR, Dr au C.R.B.t – Constantine pour leurs patience, leurs disponibilité et surtout leurs judicieux conseils, qui ont contribué à alimenter ma réflexion.

J'adresse également tous mes remerciements à Monsieur BENSEGUENI Abderrahmane, Professeur à l'université Frères Mentouri - Constantine 1, de me faire l'honneur de présider ce jury.

Je tiens aussi à remercier chaleureusement, Monsieur MOKRANI El Hassen, Maître Assistant A à l'université Frères Mentouri - Constantine 1, pour l'honneur qu'il m'a fait en acceptant d'examiner ce travail.

Je remercie également toute l'équipe du laboratoire 18- C.R.B.t – Constantine et les intervenants professionnels responsables de ma formation, pour avoir assuré la partie théorique de celle-ci.

Je remercie également toute l'équipe pédagogique de l'université de Frères Mentouri principalement les intervenants professionnels responsables de ma formation durant tous mes cycles d'étude universitaire.

Je remercie toute ma famille, mes amis pour leur soutien constant et leurs encouragements.

Dédicaces

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur

soutien et leurs

prières tout au long de mes études,

A mes chers frères, sœurs, pour leur appui et leur encouragement, et leur soutien moral

, A Mon oncle Med Cherif Badaoui

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fuit de votre

soutien infaillible,

Merci d'être toujours avec moi.

Abréviations:

AE : Encodeur Automatique

AI : Artificial Intelligence

CAS : Chemical Abstracts Service

CBOW : Continuous Bag-of-Words

CNN : Convolutional Neural Networks

DL : Deep Learning

DL50: Median lethal dose

Doc2Vec: Document to Vector

FP growth: Frequent Pattern growth

HTES : High Throughput Experimental Screening

HTVS: High Throughput Virtual Screening

IGC50:50% Inhibition Growth Concentration

MAE: Mean Absolute Error

ML : Machine learning

MLlib : Machine Learning library

MSE: Mean squared error

NLP: Natural Language Processing

NLTK: Natural Language Toolkit

OOP: Object-Oriented programming

PV-DBOW : Paragraph Vector-Distributed Bag Of Words

PV-DM : Paragraph Vector-Distributed Memory

QSAR : Relation Activité Structure Quantitative

R2 : Square Correlation Coefficient

RMSE : Mean Squared Error

RNN : Recurrent Neural Network

SG: Skip Gram

SMILES: Simplified Molecular Input Line Entry Specification

SVM: Support Vector Machine

T. pyriformis : Tetrahymena Pyriformis

Word2Vec: Word to Vecto

Table des matières

Liste des Tableaux

Table des figures :

INTRODUCTION GENERALE

Introduction générale

Le défi de la découverte des substances biologiquement actives dans le traitement, la thérapie, la prévention ou le diagnostic d'une maladie ou autre, est de trouver des molécules qui répondent à plusieurs contraintes aux profils métaboliques souhaités.

La difficulté de ce processus vient du fait que seule une petite fraction d'une base de données des millions de molécules commercialement disponibles est thérapeutiquement pertinente.

L'optimisation de ces contraintes est difficile pour les expériences biologiques à grande échelle, même s'ils sont pour la plupart automatisés, ils restent coûteux et prennent du temps.

Pour cette raison, des méthodes de calcul et des approches basées sur les informations provenant de la structure du ligand et du récepteur, telles que le criblage virtuel qui vise à identifier les hits de bibliothèques virtuelles, par des recherches basées sur la similarité ou par ancrage moléculaire. Une autre approche est la conception automatisée de nouvelles molécules aux propriétés spécifiques sont automatiquement générées par des méthodes telles que la conception de novo basée sur la structure. [1][2]

Ces approches sont encore plus difficiles lorsque nous ne disposons pas d'informations préalables. [3]

Récemment, l'intelligence artificielle, a été largement utilisée pour la conception moléculaire, car elle peut apprendre les propriétés d'exemples d'entraînement réels spécifiques, puis générer automatiquement de nouvelles entités synthétiques avec des caractéristiques similaires. Plusieurs communautés de l'industrie et du milieu universitaire ont s'orienté vers l'utilisation du Deep Learning.

Dans notre travail de mémoire, nous allons proposer une nouvelle approche inverse QSAR basé sur le Deep Learning et le traitement naturelle de texte. Notre approche suivre trois étapes dont, la première est le traitement des jeux de données SMILES.
Dans cette étape, nous générons un vocabulaire contenant tous les sous-structures de jeux de données d'entrée. Ensuite, nous allons entraîner un modèle de Deep Learning embarqué afin de prédire et transformer les données en données vectorielles numériques.
Dans la deuxième étape, nous avons entraîné un modèle de réseaux de neurones convolutifs en utilisant les vecteurs numériques générés de l'étape précédente. Cette étape, va générer une

fonction de régression pour prédire les activités biologiques. Dans la troisième étape, nous allons proposer une méthode de recherche stochastique et itérative afin de générer de novo molécule avec l'activité ciblé. Les résultats de performance montre que notre approche atteint une coefficient de détermination égale à 89% pour les données IGC50 et 85% pour les données de LD50.

En plus de l'introduction et la conclusion générales, ce manuscrit est principalement structurée en deux parties .Une partie qui décrit l'état de l'art elle regroupe le chapitres 1 et le chapitre 2. Une seconde partie consacrée à la description de la contribution et les résultats expérimentaux obtenus :

Chapitre 1 présente les principaux concepts de la chimio-informatique : QSAR.

Chapitre 2 décrit les techniques d'intelligence computationnelles utilisé dans ce travail.

La deuxième partie est destinée à notre contribution et les résultats expérimentaux obtenus.

CHAPITRE 1: CHIMIO-INFORMATIQUE

1. Introduction

Dans notre travail nous nous intéressons à l'élaboration de chimio-informatique. Pour cela, il est nécessaire de bien comprendre ce contexte. De ce fait nous consacrons ce chapitre à la présentation des Concepts de base de ce paradigme.

La Relation quantitative Structure Activité (QSAR) est le procédé par lequel une structure chimique est corrélée avec un effet bien déterminé comme l'activité biologique

La chimio-informatique est un domaine de la science qui implique l'application de l'informatique à des problèmes liés à la chimie. Elle fournit des outils et des méthodes permettent d'analyser et de traiter des données issues des différents domaines de la chimie.

Dans ce chapitre nous allons présenter les notions fondamentales sur le chimio-informatique, QSAR ainsi que quelques notions.

2. Description du Dataset

2.1. Tetrahymena Pyriformis IGC50

Pendant plus de deux décennies, de nombreuses études ont utilisé T. pyriformis pour développer des modèles linéaires QSAR. L'ensemble de données de T. pyriformis reste une excellente source primaire d'informations, en termes de taille, de diversité moléculaire et de qualité. Il est considéré comme un ensemble de données de haute qualité car il a été développé dans un seul laboratoire [8]. Jin J. Li et coll. [5] ont compilé un ensemble de données contenant des données expérimentales de toxicité aiguë de 1792 composés contre T. pyriformis en combinant les données de Li et al. [6] Et différentes références [7], [8] - [15]. La toxicité est exprimée comme la concentration qui provoque une inhibition de croissance de 50% (IGC50) après des temps très courts 40 h ou 48 h. Le log1 / GIC50 sont répertoriés dans le tableau le suivant :

Table 1-1.présentation du log1 / GIC50

Narcotic toxicants
log $1/IGC_{50}$ = 1.44(±0.04)+1.00(±0.02)log*BCF*max -0.15(±0.01) E_{LUMO}
Phenols and anilines
log $1/IGC_{50}$ = 2.17(±0.05)+0.78(±0.03)log*BCF*max -0.39(±0.03) E_{LUMO}
Narcotic amines
log $1/IGC_{50}$ = 2.05(±0.11)+0.79(±0.08)log*BCF*max
Esters
log $1/IGC_{50}$ = 1.30(±0.09)+1.05(±0.04)log*BCF*max -0.39(±0.04) E_{LUMO}
Aldehydes
log $1/IGC_{50}$ = -1.61(±1.21)+0.67(±0.06)log*BCF*max *+13.51*(±4.03) O_{DDI}
α, β- Unsaturated alcohols
log $1/IGC_{50}$ = 1.82(±0.13)+0.91(±0.09)log*BCF*max -15.97(±4.84) E_{LUMO}

2.2. Quelques statistiques

La précision des modèles de régression est caractérisée par les estimations suivantes: les intervalles de confAInce à 95% des paramètres du modèle, le coefficient de détermination (R2), l'erreur quadratique moyenne (estimation de la varAInce d'erreur, s2), la valeur F:

0 1-2 quelques satistiques sur le modéle

Toxiques narcotiques
• Coefficient de détermination R2 = 0,90,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,10,
• Valeur F = 1579,39,
• Nombre de produits chimiques, n = 351.
Phénols et anilines
• Coefficient de détermination R2 = 0,77,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,15,
• Valeur F = 574,8,
• Nombre de produits chimiques, n = 353.
Amines narcotiques
• Coefficient de détermination R2 = 0,77,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,19,
• Valeur F = 92,18,
• Nombre de produits chimiques, n = 30.
Les esters
• Coefficient de détermination R2 = 0,89,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,09,
• Valeur F = 412,7,
• Nombre de produits chimiques, n = 101.
Aldéhydes
• Coefficient de détermination R2 = 0,61,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,16,
• Valeur F = 71,90,
• Nombre de produits chimiques, n = 94.
α, β- alcools insaturés
• Coefficient de détermination R2 = 0,81,
• Erreur quadratique moyenne (estimation de la varAInce d'erreur) s2 = 0,22,
• Valeur F = 62,44,
• Nombre de produits chimiques, n = 32.

3. Chimio-informatique

La Chimio-informatique est l'utilisation de techniques informatiques et informationnelles, appliquées à une gamme de problèmes dans le domaine de la chimie. Ces techniques sont utilisées dans les sociétés pharmaceutiques dans le cadre de la chimie combinatoire.

La Chimio-informatique combine les domaines de travail scientifique de la chimie et de l'informatique en particulier dans le domaine de la théorie chimique des graphes et de l'exploitation de l'espace chimique. [4]

4. La chimie combinatoire

La chimie combinatoire (réelle ou virtuelle) un moyen pratique pour prédire et synthétiser une grande quantité de molécules en chimie pharmaceutique et agrochimique [17-19]. Elle est indispensable dans le progrès de la synthèse automatique et parallèle survenu ces 20 dernières années. Elle repose sur l'idée d'obtenir sous certaines conditions le plus grand nombre de produits possibles d'une réaction particulière [20-21]. Comme le mot l'indique, ces possessions dites «combinatoires» sont très nombreuses mais ne sont pas infinies, d'où le problème du traitement (réel ou virtuel) de ces molécules. Aux données combinatoires s'ajoutent de nouvelles molécules, issues des synthèses, des extractions et d'autres procédés chimiques, dans les bases de données chimiques à caractère académique ou industriel. Ainsi, chaque année, la base de molécules chimiques du CAS augmente de millions de nouveaux composants. Ensuite le codage et l'enregistrement des structures, des propriétés physicochimiques et biologiques de ces molécules, générant plus d'informations. L'organisation, l'analyse, la recherche et la gestion de cette grande quantité d'informations ouvrent de nouvelles possibilités aux techniques novatrices de chimie informatique, parmi les comptes :

HTES: méthode en chimie médicinale de sélection de nouvelles molécules têtes de série ou candidats médicaments

HTVS: un procédé in silico permettant de filtrer et de donner un score à des molécules d'une bibliothèque en fonction de leur affinité prédite avec une cible biologique. Ainsi, il permet de prédire l'activité des petites molécules organiques présentes dans des grandes banques de

données publiques ou privées et de focaliser l'approche expérimentale sur les molécules candidates les plus prometteuses.

La fouille de données (data-mining) : elle désigne l'analyse de données depuis différentes perspectives et de transformer ces données en informations utiles, en établissant des relations entre les données ou en repérant des patterns. [16]

5. Molécule similaire

Le concept de similarité, où les molécules peuvent être groupées en fonction de leurs effets biologiques ou de leurs propriétés physicochimiques, a été largement utilisé dans la découverte de médicaments. Certaines zones d'intérêt particulier ont été dans la découverte de plomb et l'optimisation des composés.

Le principe de propriété similaire déclare que les molécules structurellement similaires ont tendance avoir des propriétés similaires.

Les méthodes de similarité moléculaire peuvent être grossièrement classées en méthodes de similarité bidimensionnelles (2D) et tridimensionnelles (3D). Typiquement, les méthodes de similarité 2D utilisent des empreintes digitales. [22][23]

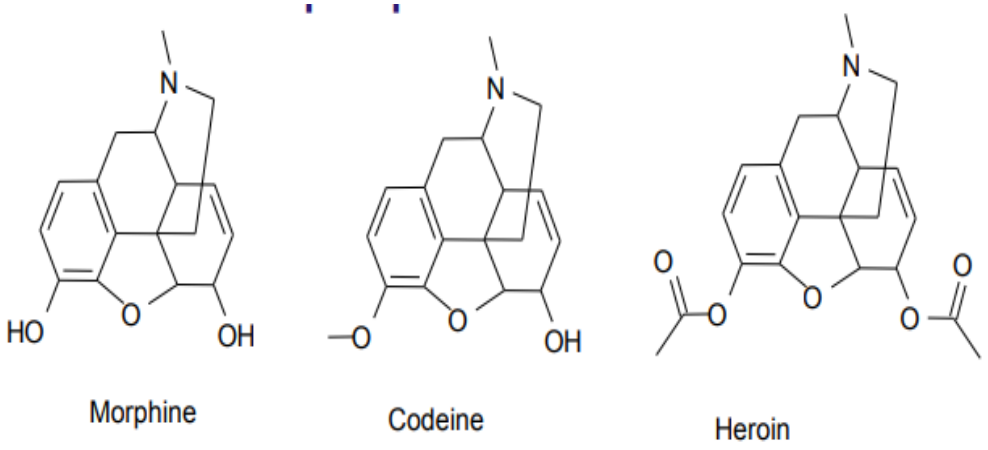


Figure 1. 1 exemple de la similarité

5.1. L'empreinte digitale

Les empreintes moléculaires sont un moyen de coder la structure d'une molécule. Le type d'empreinte le plus commun est une série de chiffres binaires (bits) qui représentent la présence ou l'absence de sous-structures particulières dans la molécule. La comparaison des empreintes digitales vous permet de déterminer la similarité entre deux molécules, de trouver des correspondances à une sous-structure de requête, etc.

6. Relation activité-structure quantitative (QSAR)

C'est le processus par lequel la structure chimique est corrélée quantitativement avec un processus bien défini, tel que la réactivité chimique ou l'activité biologique [21]

La concentration d'une substance requise exprimée quantitativement L'activité biologique afin de donner une certaine réponse biologique. L'expression mathématique peut être alors utilisée pour la prédiction de la réponse biologique d'autres structures chimiques [24].

Des propriétés telles que la lipophilicite, la solubilité et la perméabilité peuvent exprimer La réactivité chimique. [25]. La forme mathématique QSAR est : [26]

$$\text{Activité} = f(x)$$

X : propriétés physico-chimiques et / ou propriétés structurelles

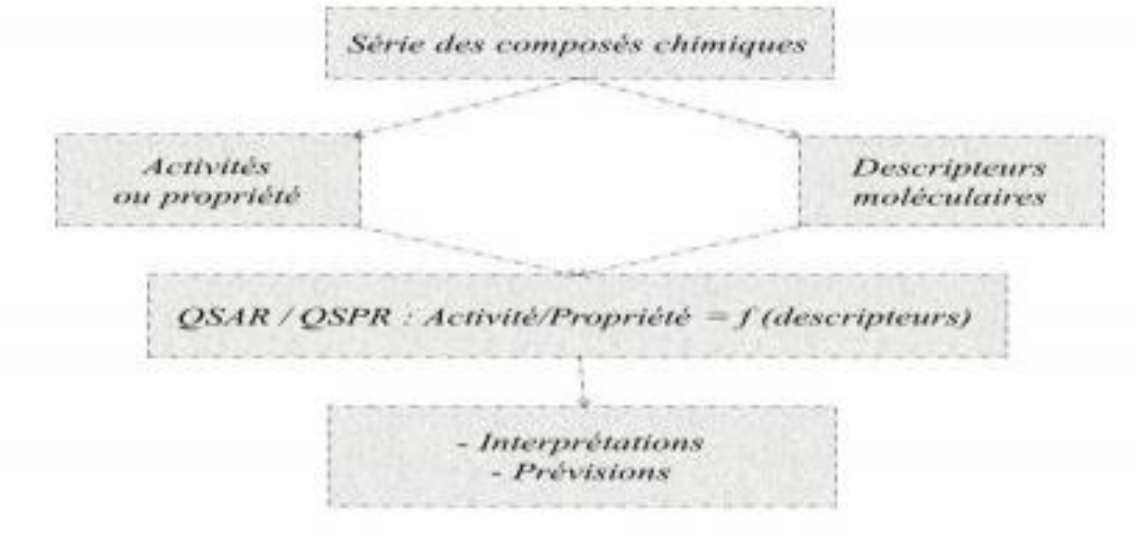


Figure 1. 2 Relation activité-structure quantitative (QSAR) [26]

6.1. Utilisation de QSAR

Parmi et les raisons les plus importantes d'utilisation du QSAR, on distingue :

- Testes de toutes les substances chimiques en termes de propriétés toxicologiques et environnementales avant leur utilisation, ce qui gâchera beaucoup de temps et d'argent.
- le test des substances chimiques est très court car Le modèle QSAR est très rapide
- réduction de nombre d'animaux utilisés dans les expériences par Le modèle QSAR

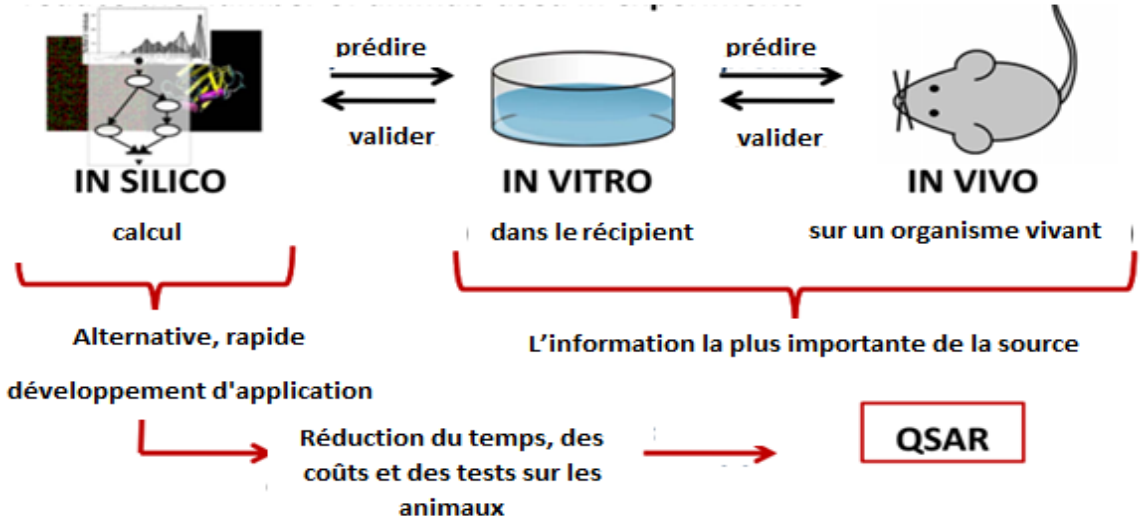


Figure 1. 3 Schéma de raison de besoins de modelé QSAR

6.2. Méthode d'analyse les donnes

Différentes méthodes statistiques ont été utilisées dans QSAR pour l'extraction d'informations utiles à partir des données.

Problèmes de régression:

- La régression linéaire multiple
- Moindres carrés partiels
- Réseau de neurones de propagation en retour
- Réseau de neurones de régression générale
- Problèmes de classification:

- Analyse discriminante linéaire
- Régression logistique
- Arbre de décision
- K-plus proche voisin
- Réseau neuronal probabiliste
- Machine de vecteur de soutien

Méthodes récemment développées:

- Noyau partiel des moindres carrés
- Régression robuste du continuum
- Régression paresseuse locale
- Nombre d'intervalles flous k-plus proche voisin
- Plan de projection rapide classifie

6.3. Outils et technique de QSAR

Habituellement, on exprime Les données biologiques sur une échelle logarithmique en raison de la relation linéaire entre la réponse et le logarithme de dose dans la région centrale de la courbe de log dose-réponse. Les logarithmes inverses de l'activité (log 1/C) sont également utilisés afin d'obtenir des valeurs mathématiques plus élevées lorsque les structures sont biologiquement très efficaces. Des exemples de données biochimiques ou biologiques, utilisés dans l'analyse de QSAR, sont décrits dans le tableau suivant. [27]

Table 1-3 Types de données biologiques utilisées dans l'analyse QSAR [27]

Source d'activité	Paramètres biologiques
❖ Récepteurs isolés	
Constante de vitesse	Log *k*
Constante de Michaelis-Menten	Log $1/K_m$
Constante d'inhibition	Log $1/K_i$
❖ Systèmes cellulaires	
Constante d'inhibition	Log $1/IC_{50}$
Résistance croisée	Log CR
Données biologiques *in vitro*	Log 1/C
Mutation de gène	Log TA_{98}
❖ Systèmes in vivo	
Facteur de bioconcentration	Log BCF
Vitesses de la réaction *in vivo*	Log *I* (induction)
Vitesses pharmacodynamiques	Log *T* (clairance totale)

6.3.1. Descripteur moléculaire

Un descripteur moléculaire est considéré comme le résultat d'un processus logique et mathématique, appliqué à l'information chimique codifiée à travers la représentation d'une molécule. [28]

L'information codée d'un descripteur moléculaire dépend du type de représentation moléculaire employée et de l'algorithme défini pour son calcul. Il existe :

- Des descripteurs moléculaires simples dérivés du nombre d'atome-type ou de fragments Structuraux de la molécule.
- descripteurs2D.
- descripteurs 3D.

6.3.2. Les techniques de QSAR

Certain projet de recherche mette en jeu des données biologiques et physico-chimiques appropriées. Ces données peuvent être représentées et analysées de diverses manières. Le groupement et la classification des composés, basés sur leurs activités, sont les éléments principaux lors d'études de similarité moléculaire. Les méthodes linéaires et non linéaires sont généralement utiles pour rationaliser les relations structure-activité.

Ces deux types d'études sont nommés analyse de données statistiques multi varAIbles, ou étude de QSAR. [29]

7. Conclusion

Nous avons consacré ce chapitre à la présentation de la notion sur notre travail, Cette présentation nous a permis d'avoir les notions générales de notre travail et nous a poussé à exposer un certain nombre de points toujours en vue de compléter une vision générale sur notre contexte de travail. Ce qui nous conduit au sujet du chapitre suivant : intelligence artificielle et traitement de langage naturel.

CHAPITRE 2 : INTELLIGENCE ARTIFICIELLE ET TRAITEMENT DE LANGAGE NATUREL

1. Introduction

L'Intelligence Artificielle (AI) est un terme qui désigne la capacité d'un ordinateur ou d'une machine à accomplir des tâches ou à prendre des décisions, tout comme les humains. Le terme d'intelligence artificielle a été inventé par John McCarthy en 1956. L'AI est la branche de l'informatique qui s'occupe de l'étude et de la conception d'agents intelligents qui perçoivent son environnement et entreprend des actions qui maximisent ses chances de succès. Mais l'AI doit inclure l'apprentissage de l'expérience passée, le raisonnement pour la prise de décision, le pouvoir d'inférence et une réponse rapide. De plus, elle doit être capable de prendre des décisions sur la base des priorités et de lutter contre la complexité et l'ambiguïté On dit que les machines programmées pour effectuer des tâches, lorsqu'elles sont exécutées par des humains, nécessiteraient de l'intelligence, possèdent une intelligence artificielle. Le but scientifique de l'AI est de comprendre l'intelligence en créant des programmes informatiques qui présentent un comportement intelligent en utilisant l'inférence symbolique ou en raisonnant à l'intérieur de la machine. La définition de l'AI n'est pas indépendante du temps. Elle donne le jugement de n'importe quel système en gardant le temps à l'esprit. [30]

Dans ce chapitre, il était considéré de présenter les principaux concepts de l'AI et l'apprentissage automatique ainsi que les principaux modèles de deep learning. Ensuite, nous allons discuter les algorithmes de tendance de traitement de texte naturelle.

2. Intelligence Artificielle

L'AI peut être définie comme: «L'intelligence artificielle (AI) fait référence à la simulation de l'intelligence humaine dans des machines programmées pour penser comme des humains et imiter leurs actions. Le terme peut également être appliqué à toute machine présentant des traits associés à un esprit humain tels que l'apprentissage et la résolution de problèmes. ». Les concepteurs d'AI visent à reproduire les attributs humains tels que la créativité, le raisonnement logique et l'acquisition de connaissances dans des systèmes à différents niveaux. Les assistants virtuels et les chatbots dans les sites de réservation de voyages sont une démonstration claire de

la façon dont l'AI peut automatiser des tâches spécifiques que seuls les humains pouvaient effectuer dans le passé. [31]

2.1. Apprentissage automatique :(machine learning)

Dans les termes les plus simples, l'apprentissage automatique (ML) est un sous-ensemble de l'AI. Son cœur réside dans l'idée que les systèmes informatiques peuvent apprendre par eux-mêmes à partir des données obtenues lors de l'exécution de tâches précédentes et d'expériences passées. Cela signifie que vous n'avez pas besoin de préprogrammer un appareil AI chaque fois que vous en avez besoin pour un travail

Le ML comprend trois sous-catégories: supervisé, non supervisé et de renforcement. L'apprentissage supervisé se produit lorsqu'un système d'AI parvient à une résultat prévisible basée sur des données existantes. L'apprentissage non supervisé, quant à lui, a lieu lorsque l'agent d'AI produit un résultat imprévisible, ce pour quoi il n'a pas été préformé. Enfin, l'apprentissage par renforcement (également connu comme OOP) consiste à entraîner l'algorithme d'AI à reconnaître les récompenses et les punitions afin qu'il puisse trouver la meilleure solution à un problème. [31]

2.1.1. Types des algorithmes d'Apprentissage automatique

Les algorithmes d'apprentissage automatique sont divisés en catégories en fonction de leur objectif. Les principales catégories sont

2.1.1.1. Les algorithmes d'apprentissage supervisé

Sont un sous-ensemble de la famille des algorithmes d'apprentissage automatique qui sont principalement utilisés dans la modélisation prédictive. Un modèle prédictif est essentiellement un modèle construit à partir d'un algorithme d'apprentissage automatique et de fonctionnalités ou d'attributs à partir de données d'apprentissage, de sorte que nous pouvons prédire une valeur à l'aide des autres valeurs obtenues à partir des données d'entrée. Les algorithmes d'apprentissage supervisé tentent de modéliser les relations et les dépendances entre la sortie de prédiction cible et les caractéristiques d'entrée de sorte que nous puissions prédire les valeurs de sortie des nouvelles données en fonction des relations apprises à partir des ensembles de données précédents. Les principaux types d'algorithmes d'apprentissage supervisé comprennent:

2.1.1.2. Algorithmes de classification

Ces algorithmes créent des modèles prédictifs à partir de données d'entraînement qui ont des fonctionnalités et des étiquettes de classe. Ces modèles prédictifs utilisent à leur tour les fonctionnalités apprises à partir des données d'entraînement sur de nouvelles données auparavant invisibles pour prédire leurs étiquettes de classe. Les classes de sortie sont discrètes. Les types d'algorithmes de classification incluent les arbres de décision, les forêts aléatoires, les machines vectorielles de support (SVM) et bien d'autres. [32]

2.1.1.3. Algorithmes de régression

Ces algorithmes sont utilisés pour prédire les valeurs de sortie en fonction de certaines caractéristiques d'entrée obtenues à partir des données. Pour ce faire, l'algorithme crée un modèle basé sur les caractéristiques et les valeurs de sortie des données d'apprentissage et ce modèle est utilisé pour prédire les valeurs des nouvelles données. Les valeurs de sortie dans ce cas sont continués et non discrètes. Les types d'algorithmes de régression comprennent la régression linéaire, la régression multi-variée, les arbres de régression et la régression au lasso, entre autres.

Certaines applications de l'apprentissage supervisé sont la reconnaissance vocale, la notation de crédit, l'imagerie médicale et les moteurs de recherche. [32]

2.1.1.4. Apprentissage non supervisé

Sont la famille d'algorithmes d'apprentissage automatique qui sont principalement utilisés dans la détection de modèles et la modélisation descriptive. Cependant, il n'y a pas de catégories de sortie ou d'étiquettes basées sur lesquelles l'algorithme peut essayer de modéliser des relations. Ces algorithmes essaient d'utiliser des techniques sur les données d'entrée pour extraire les règles, détecter les modèles, et synthétiser et regrouper les points de données qui aident à dériver des informations significatives et à mieux décrire les données aux utilisateurs.

Les principaux types d'algorithmes d'apprentissage non supervisés comprennent :

2.1.1.5. Algorithmes de Clustering

L'objectif principal de ces algorithmes est de regrouper ou de grouper des points de données d'entrée dans différentes classes ou catégories en utilisant uniquement les fonctions dérivées des données d'entrée seules et aucune autre information externe. Contrairement à la classification, les

étiquettes de sortie ne sont pas connues auparavant dans le clustering. Il existe différentes approches pour construire des modèles de clustering, comme l'utilisation de moyens, de médoïdes, de hiérarchies et bien d'autres. Certains algorithmes de regroupement populaires incluent k-means, k-medoids et clustering hiérarchique. [32]

2.1.1.6. Algorithmes d'apprentissage des règles d'association

Ces algorithmes sont utilisés pour exploiter et extraire des règles et des modèles à partir d'ensembles de données. Ces règles expliquent les relations entre différentes variables et attributs et décrivent également les ensembles d'éléments fréquents et les modèles qui se produisent dans les données. Ces règles à leur tour aident à découvrir des informations utiles pour toute entreprise ou organisation à partir de leurs énormes référentiels de données. Les algorithmes populaires incluent Apriori et FP Growth. [32]

2.1.1.7. Apprentissage semi-supervisé

Dans les deux types précédents, soit il n'y a pas d'étiquettes pour toutes les observations de l'ensemble de données, soit des étiquettes sont présentes pour toutes les observations. L'apprentissage semi-supervisé se situe entre ces deux éléments. Dans de nombreuses situations pratiques, le coût de l'étiquetage est assez élevé, car cela nécessite des experts humains qualifiés pour le faire. Ainsi, en l'absence d'étiquettes dans la majorité des observations mais présentes dans peu, les algorithmes semi-supervisés sont les meilleurs candidats pour la construction du modèle. Ces méthodes exploitent l'idée que même si les appartenances aux groupes des données non étiquetées sont inconnues, ces données contiennent des informations importantes sur les paramètres du groupe.[32]

2.1.1.8. Apprentissage par renforcement

vise à utiliser les observations recueillies à partir de l'interaction avec l'environnement pour prendre des mesures qui maximiseraient la récompense ou minimiseraient le risque. L'algorithme d'apprentissage par renforcement (appelé l'agent) apprend continuellement de l'environnement de manière itérative. Dans le processus, l'agent apprend de ses expériences de l'environnement jusqu'à ce qu'il explore la gamme complète des états possibles.

Afin de produire des programmes intelligents (également appelés agents), l'apprentissage par renforcement passe par les étapes suivantes:

- L'état d'entrée est observé par l'agent.

- ❖ La fonction de prise de décision est utilisée pour obliger l'agent à effectuer une action.
- ❖ Une fois l'action effectuée, l'agent reçoit une récompense ou un renforcement de l'environnement.
- ❖ Les informations sur la paire état-action concernant la récompense sont stockées.

Certaines applications des algorithmes d'apprentissage par renforcement sont les jeux de société (échecs, go), les mains robotiques et les voitures autonomes.[32]

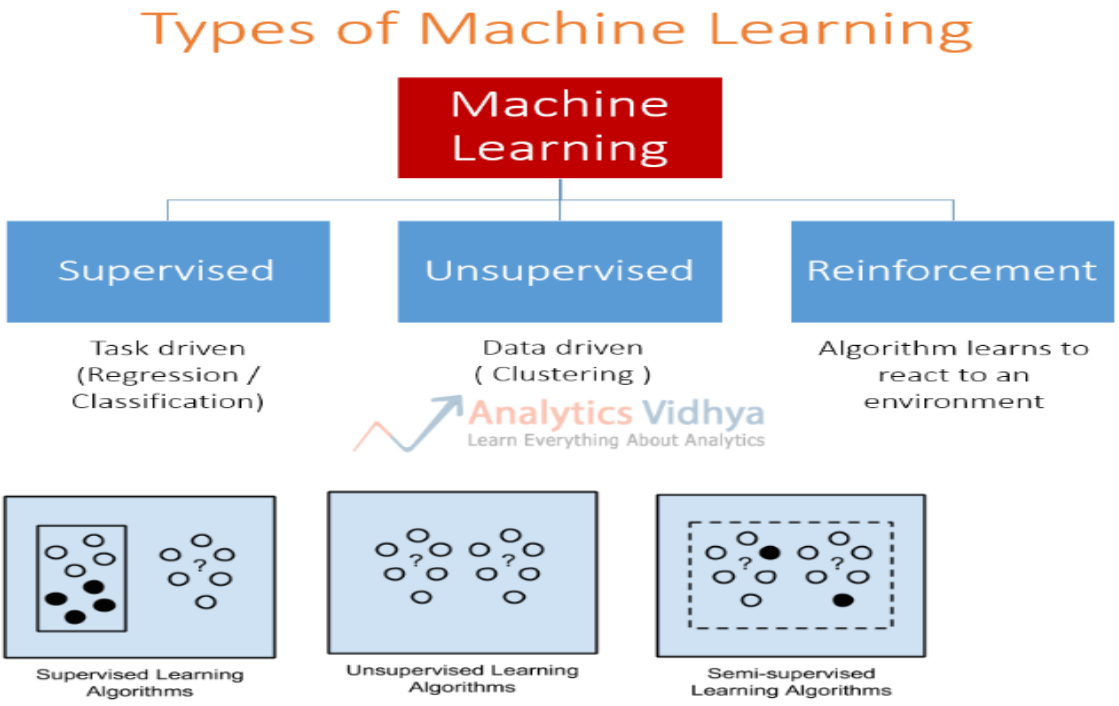


Figure 2. 1 types of machine learning [32]

2.2. Deep Learning

2.2.1. Définition

Deep learning est un ensemble d'algorithmes de ML qui tente d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il utilise généralement les réseaux de neurones artificiels. Les niveaux de ces modèles statistiques appris correspondent à des niveaux de concepts distincts, où les concepts de niveau supérieur sont définis à partir de ceux de niveau

inférieur, et les mêmes concepts de niveau inférieur peuvent aider à définir de nombreux concepts de niveau supérieur. [33]

2.2.2. Domain d'application de Deep Learning

Voici quelques domaines :

- Voitures autonomes
- Agrégation de nouvelles et détection de nouvelles de fraude
- Traitement du langage naturel
- Assistants virtuels
- Divertissement
- Reconnaissance visuelle
- Détection de fraude
- Soins de santé
- Personnalisations
- Détection du retard de développement chez les enfants
- Coloration des images en noir et blanc
- Ajout de sons à des films muets
- Traduction automatique
- Génération d'écriture automatique
- Jeu automatique
- Traductions linguistiques
- Restauration des pixels
- Descriptions des photos
- Prévisions démographiques et électorales
- Rêver profond [34]

2.3. Les réseaux de neurones artificiels

2.3.1. Introduction

Vous êtes-vous déjà demandé comment Facebook sait comment suggérer le bon ami à taguer? En parlant de cela, comment fonctionne l'algorithme de recherche d'images de Google? Oui, vous avez raison, il y a un réseau de neurones impliqué dans toutes ces tâches. Pour être plus précis, nous parlons de réseaux de neurones convolutionnels. Même si cela ressemble à un

étrange mélange de biologie et d'informatique (tout ce qui concerne les réseaux de neurones sonne un peu comme ça), c'est un mécanisme très efficace utilisé pour la reconnaissance d'image. Bien sûr, il est motivé par les systèmes biologiques et le fonctionnement du cerveau, en particulier le cortex visuel.[35]

2.3.2. La biologie derrière l'idée

Les neurones individuels du cortex visuel ne répondent aux stimuli que dans une région restreinte du champ visuel connue sous le nom de champ récepteur. S'il y a une certaine fonctionnalité dans notre champ visuel, des neurones spécifiques seront activés et d'autres non. Cela a été prouvé par une expérience fascinante réalisée par Hubel et Wiesel en 1962

Alors, comment les réseaux de neurones convolutionnels l'utilisent-ils pour la reconnaissance d'image? Eh bien, ils utilisent cette idée pour différencier les images données et déterminer les caractéristiques uniques qui font d'un avion un avion ou un serpent - un serpent. Ce processus se produit dans notre esprit inconsciemment. Par exemple, lorsque nous regardons l'image d'un avion, nous pouvons l'identifier comme un avion en distinguant des caractéristiques telles que deux ailes, un conte, des fenêtres, etc. Les réseaux de neurones convolutifs font la même chose, mais ils détectent d'abord un niveau inférieur des fonctionnalités telles que des courbes et des arêtes, puis ils le construisent en concepts plus abstraits. [35]

2.3.3. Réseaux de neurones convolutifs

Un réseau neuronal convolutif, ou CNN, est un réseau neuronal d'apprentissage en profondeur conçu pour traiter des tableaux structurés de données telles que des images. Les réseaux de neurones convolutifs sont largement utilisés dans la vision par ordinateur et sont devenus l'état de l'art pour de nombreuses applications visuelles telles que la classification d'images, et ont également trouvé du succès dans le traitement du langage naturel pour la classification de texte..[36]

2.3.3.1. Structure des réseaux de neurones convolutifs

Afin d'atteindre la fonctionnalité dont nous avons parlé, le réseau de neurones convolutifs traite l'image à travers plusieurs couches. Nous les examinerons en détail dans les prochains chapitres de cet article, mais pour l'instant, faisons simplement un aperçu d'eux et de leurs objectifs:

- Couche convolutive - Utilisée pour détecter les fonctionnalités

- Couche de non-linéarité - Introduction de la non-linéarité au système
- Couche de regroupement (sous-échantillonnage) - Réduit le nombre de poids et contrôle le sur-ajustement
- Flattening Layer - Prépare les données pour le réseau neuronal classique
- Couche entièrement connectée - Réseau neuronal standard utilisé pour la classification

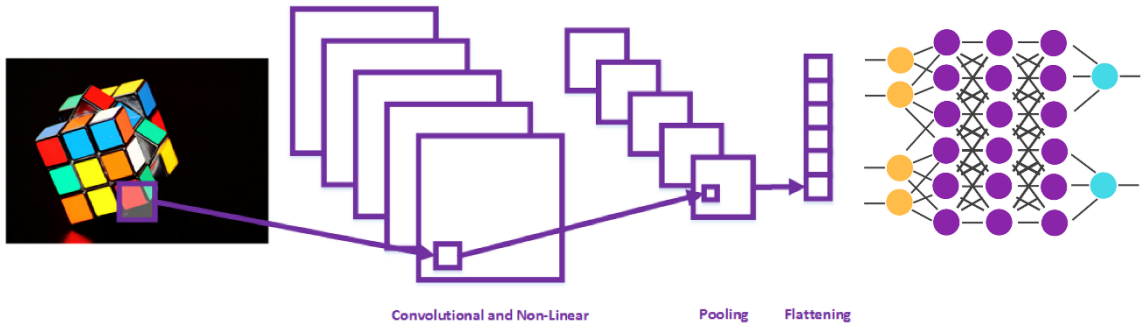


Figure 2. 2 Schéma d'un réseau CNN [35]

2.3.3.2. Encodeur automatique (Auto-encoder)

Un auto-encodeur (AE) est une technique d'apprentissage non supervisée pour les réseaux de neurones qui apprend des représentations de données efficaces (encodage) en apprenant au réseau à ignorer le «bruit» du signal.

Le réseau d'auto-encodeur a trois couches: l'entrée, une couche cachée pour l'encodage et la couche de décodage de sortie. En utilisant la rétro-propagation, l'algorithme non supervisé s'entraîne en continu en définissant les valeurs de sortie cibles pour égaler les entrées. Cela force la plus petite couche de codage caché à utiliser la réduction dimensionnelle pour éliminer le bruit et reconstruire les entrées. [38]

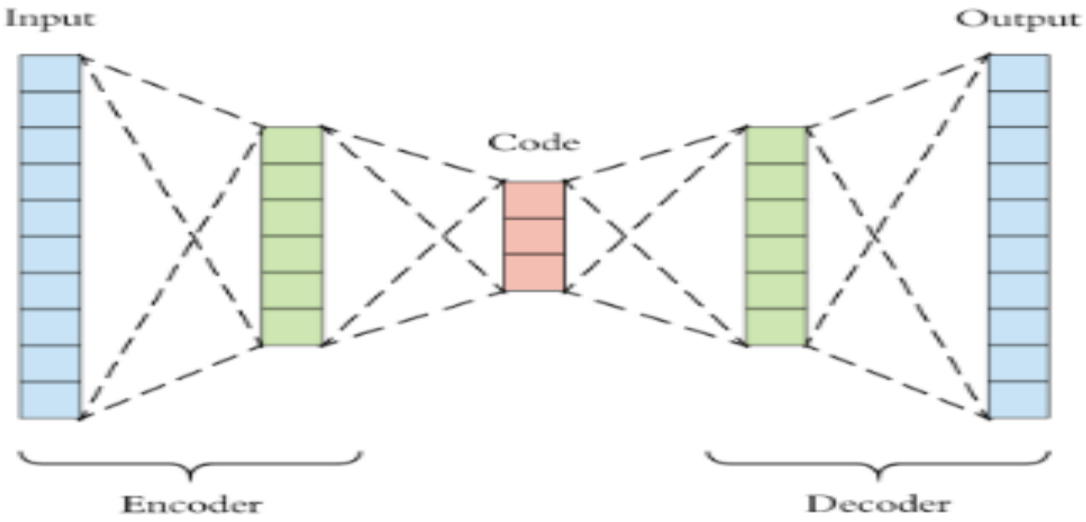


Figure 2. 3 Schéma d'un réseau Auto-encodeur [37]

2.3.3.3. Types d'Auto-encoder [38]

1. auto-encodeur de dé-bruitage
2. auto-encodeur clairsemé
3. auto-encodeur vibrationnel (VAE)
4. auto-encodeur Contractive (CAE)

2.3.4. Réseau neuronal récurrent (RNN)

Un réseau neuronal récurrent est un type de réseau neuronal qui contient des boucles, permettant de stocker des informations au sein du réseau. En bref, les réseaux neuronaux récurrents utilisent leur raisonnement des expériences précédentes pour informer les événements à venir. Les modèles récurrents sont précieux dans leur capacité à séquencer les vecteurs, ce qui ouvre l'API à l'exécution de tâches plus complexes.[39]

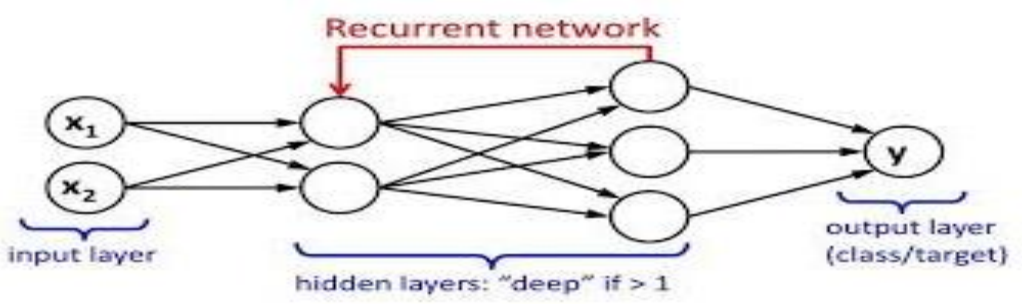


Figure 2. 4 Schéma d'un RNN [39]

2.3.4.1. Fonctionnement des réseaux de neurones récurrents

Les réseaux de neurones récurrents peuvent être considérés comme une série de réseaux reliés entre eux. Ils ont souvent une architecture en forme de chaîne, ce qui les rend applicables à des tâches telles que la reconnaissance vocale, la traduction de la langue, etc. Un RNN peut être conçu pour fonctionner sur des séquences de vecteurs en entrée, en sortie ou les deux. Par exemple, une entrée séquencée peut prendre une phrase comme entrée et générer une valeur de sentiment positive ou négative. En variante, une sortie séquencée peut prendre une image comme entrée et produire une phrase comme sortie.[39]

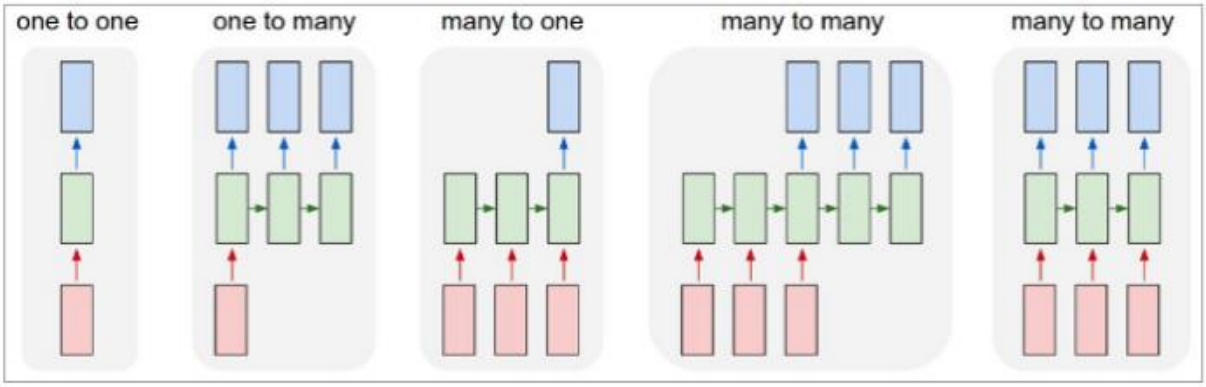


Figure 2. 5 Image présentatrice des types entré-sortie RNN [39]

Imaginons former un RNN au mot «happy», étant donné les lettres «h, a, p, y». Le RNN sera formé sur quatre exemples distincts, chacun correspondant à la probabilité que les lettres tombent dans une séquence voulue. Par exemple, le réseau sera formé pour comprendre la probabilité que la lettre «a» suive dans le contexte de «h». De même, la lettre «p» doit apparaître après les séquences de «ha». Encore une fois, une probabilité sera calculée pour la lettre «p» après la

séquence «hap». Le processus se poursuivra jusqu'à ce que les probabilités soient calculées pour déterminer la probabilité que les lettres tombent dans la séquence prévue. Ainsi, lorsque le réseau reçoit chaque entrée, il déterminera la probabilité de la lettre suivante en fonction de la probabilité de la lettre ou de la séquence précédente. Au fil du temps, le réseau peut être mis à jour pour produire des résultats plus précisément. [39]

2.3.4.2. Applications des réseaux de neurones récurrents

Un exemple courant de réseaux neuronaux récurrents est la traduction automatique. Par exemple, un réseau de neurones peut prendre une phrase d'entrée en espagnol et la traduire en une phrase en anglais. Le réseau détermine la probabilité de chaque mot dans la phrase de sortie sur la base du mot lui-même et de la séquence de sortie précédente.[39]

3. Le traitement du langage naturel

Le traitement du langage naturel (NLP) est la capacité d'un programme informatique à comprendre le langage humain tel qu'il est parlé. La NLP est une composante de l'intelligence artificielle (AI).

Le développement d'applications NLP est difficile car les ordinateurs exigent traditionnellement que les humains leur «parlent» dans un langage de programmation précis, sans ambiguïté et hautement structuré, ou à travers un nombre limité de commandes vocales clairement énoncées. Le discours humain, cependant, n'est pas toujours précis - il est souvent ambigu et la structure linguistique peut dépendre de nombreuses variables complexes, notamment l'argot, les dialectes régionaux et le contexte social.[40]

3.1. Comment fonctionne le traitement du langage naturel: techniques et outils

La syntaxe et l'analyse sémantique sont deux techniques principales utilisées avec le traitement du langage naturel. La syntaxe est la disposition des mots dans une phrase pour donner un sens grammatical. La NLP utilise la syntaxe pour évaluer le sens d'une langue basée sur des règles grammaticales. Les techniques de syntaxe utilisées comprennent l'analyse syntaxique (analyse grammaticale d'une phrase), la segmentation de mots (qui divise un grand morceau de texte en unités), la rupture de phrases (qui place les limites de la phrase dans de grands textes), la segmentation morphologique (qui divise les mots en groupes) et la racine (qui divise les mots avec une inflexion en eux pour former des racines).

La sémantique implique l'utilisation et la signification des mots. La NLP applique des algorithmes pour comprendre le sens et la structure des phrases. Les techniques utilisées par la NLP avec la sémantique incluent la désambiguïsation du sens des mots (qui dérive le sens d'un mot en fonction du contexte), la reconnaissance d'entités nommées (qui détermine les mots pouvant être classés en groupes) et la génération de langage naturel (qui utilisera une base de données pour déterminer la sémantique derrière les mots).

Les approches actuelles de la NLP sont basées sur le deep learning . Les modèles de deep learning nécessitent d'énormes quantités de données étiquetées pour s'entraîner et identifier les corrélations pertinentes, et l'assemblage de ce type d'ensemble de données volumineuses est actuellement l'un des principaux obstacles à la NLP.

Les approches antérieures de la NLP impliquaient une approche plus basée sur des règles, dans laquelle des algorithmes d'apprentissage automatique plus simples étaient informés des mots et des phrases à rechercher dans le texte et recevaient des réponses spécifiques lorsque ces phrases apparaissaient. Mais l'apprentissage en profondeur est une approche plus flexible et intuitive dans laquelle les algorithmes apprennent à identifier l'intention des locuteurs à partir de nombreux exemples, presque comme la façon dont un enfant apprendrait le langage humain.

Trois outils couramment utilisés pour la NLP sont NLTK, Gensim et Intel NLP Architect. NTLK, Natural Language Toolkit, est un module python open source avec des ensembles de données et des didacticiels. Gensim est une bibliothèque Python pour la modélisation de sujets et l'indexation de documents. Intel NLP Architect est également une autre bibliothèque Python pour les topologies et techniques d'apprentissage en profondeur.[40]

3.2. Les intégrations de mots (Word embidding)

Les intégrations de mots sont un type de représentation de mots qui stocke les informations contextuelles dans un vecteur de faible dimension. Cette approche a acquis une popularité extrême avec l'introduction de Word2Vec en 2013, un groupe de modèles permettant d'apprendre l'intégration de mot d'une manière efficace en termes de calcul. Et Doc2Vec peut être vu une extension de Word2Vec dont le but est de créer un vecteur de représentation d'un document.

Word2Vec et Doc2Vec sont implémentés dans plusieurs packages / bibliothèques. le package gensim implémentait à la fois Word2Vec et Doc2Vec. La bibliothèque de machine learning de

Google tensorflow fournit la fonctionnalité Word2Vec. En outre, la bibliothèque MLlib de spark implémente également Word2Vec. [41]

3.2.1. Word2vec

Deux algorithmes les plus populaires pour créer des représentations Word2Vec sont le modèle ‘’Skip-Gram’’ et le modèle ‘’Continuous Bag-of-Words CBOW’’. Passons aux détails de ces deux algorithmes.

Word2Vec utilise un simple réseau de neurones avec une seule couche cachée pour apprendre les poids. Différents de la plupart des autres modèles d'apprentissage automatique, nous ne sommes pas intéressés par les prédictions que ce réseau de neurones pourrait faire. Au lieu de cela, nous ne nous soucions que des poids de la couche cachée, car ces poids sont en fait le mot intégration / vecteurs que nous sommes sur le point d'apprendre.[41]

3.2.1.1. Skip-Gram

Pour le modèle Skip-Gram, la tâche du réseau de neurones simple est la suivante:

Étant donné un mot d'entrée dans une phrase, le réseau prédira la probabilité que chaque mot du vocabulaire soit le mot voisin de ce mot d'entrée.

Les exemples de formation (train) du réseau neuronal sont des paires de mots qui se composent du mot d'entrée et de ses mots voisins. Par exemple, considérez la phrase «He says make America great again». Et une taille de fenêtre de 2 colonnes. Les exemples de formation sont:

Table 2. 1 un exemple d'entrainement [41]

Sentence	Training examples
He says make America great again.	(he,says), (he,make)
He **says** make America great again.	(says,he), (says,make), (says,america)
He says **make** America great again.	(make,he), (make,says), (make,america), (make,great)
He says make **America** great again.	(america,says),(america,make) (america,great),(america,again)
He says make America **great** again.	(great,make), (great,america),(great,again)
He says make America great **again**.	(again,america), (again,great)

Pour que les exemples soient entraînés par le réseau neuronal, nous devons représenter les mots sous une forme numérique. Nous utilisons des vecteurs one-hot, dans lesquels la position du mot d'entrée est «1» et toutes les autres positions sont «0». Ainsi, les entrées du réseau de neurones ne sont que des vecteurs d'entrée unique, et la sortie est également un vecteur avec la dimension du vecteur unique, contenant, pour chaque mot du vocabulaire, la probabilité qu'un mot voisin sélectionné au hasard soit ce mot de vocabulaire.

Examinons maintenant l'architecture du réseau neuronal. Par exemple, supposons que nous utilisons un vocabulaire de taille V et une couche cachée de taille N, le diagramme suivant montre l'architecture du réseau:

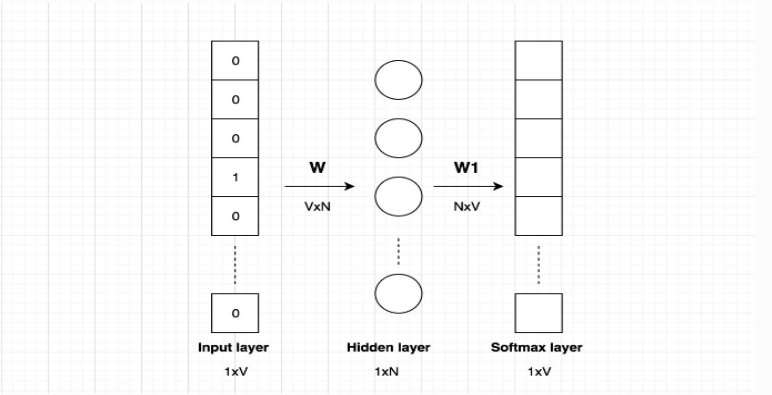


Figure 2. 6 Diagramme montre l'architecture du réseau neuronal [41]

L'entrée est un vecteur one-hot avec une dimension 1xV. La dimension de la matrice de poids du calque caché est VxN. Si nous les multiplions, nous obtiendrons un vecteur de dimension 1xN.

$$[0\ 0\ 0\ 0\ 1\ 0] \times \begin{bmatrix} 10 & 23 & 15 \\ 3 & 14 & 9 \\ 18 & 26 & 2 \\ 10 & 17 & 7 \\ 12 & 23 & 8 \\ 9 & 10 & 12 \end{bmatrix} = [12\ 23\ 8]$$

Figure 2. 7 Calcule du vecteur [41]

Si vous regardez vraiment le calcul de la matrice ci-dessus, vous pouvez voir que la matrice de poids de la couche cachée fonctionne en fait comme une table de recherche, elle sélectionnera simplement la ligne de matrice correspondant au «1». La sortie du calcul matriciel est le mot d'intégration / vecteur pour le mot d'entrée. Il y a V lignes dans la matrice de poids, chaque ligne correspondant à un vecteur de mot dans le vocabulaire. C'est pourquoi nous nous intéressons uniquement à l'apprentissage de la matrice de poids de la couche cachée et nous l'appelons les intégrations de mots.

La couche de sortie est une couche softmax de dimension 1xV, chaque élément correspondant à la probabilité que ce mot soit le mot que vous sélectionnez au hasard à proximité du mot d'entrée. [41]

3.2.1.2. CBOW

Le modèle CBOW (Continuous Bag-of-Words) est juste l'opposé de Skip-Gram.

Pour le modèle CBOW, la tâche du réseau de neurones simple est la suivante: Étant donné un contexte de mots (entourant un mot) dans une phrase, le réseau prédira la probabilité que chaque mot du vocabulaire soit le mot.[41]

3.2.2. Doc2Vec

Le concept de Doc2Vec est en fait assez simple, si vous êtes déjà familier avec le modèle Word2vec. Le modèle Doc2vec est basé sur Word2Vec, avec seulement l'ajout d'un autre vecteur (ID de paragraphe) à l'entrée. L'architecture du modèle Doc2Vec est illustrée ci-dessous:

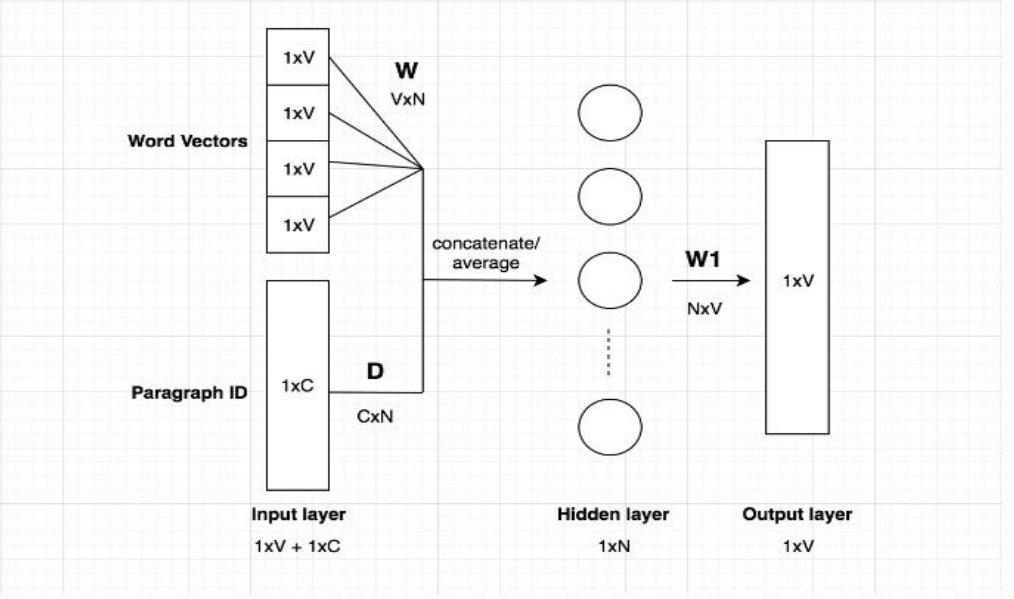


Figure 2. 8 L'architecture du model doc2vec [41]

Le diagramme ci-dessus est basé sur le modèle CBOW, mais au lieu d'utiliser uniquement des mots proches pour prédire le mot, nous avons également ajouté un autre vecteur de caractéristiques, qui est unique au document. Ainsi, lors de l'apprentissage des vecteurs de mots W, le vecteur de document D est également entraîné et à la fin de l'apprentissage, il contient une représentation numérique du document.

Les entrées sont constituées de vecteurs de mots et de vecteurs d'ID de document. Le mot vecteur est un vecteur one-hot avec une dimension 1xV. Le vecteur d'ID de document a une dimension de 1xC, où C est le nombre total de documents. La dimension de la matrice de poids W de la couche cachée est VxN. La dimension de la matrice de poids D de la couche cachée est CxN.

Le modèle ci-dessus est appelé version à mémoire distribuée de Paragraph Vector (PV-DM). Un autre algorithme Doc2Vec basé sur Skip-Gram est appelé la version Distributed Bag of Words de Paragraph Vector (PV-DBOW).[41]

3.3. Avantages de la NLP

La NLP héberge des avantages tels que: [42]

- Amélioration de la précision et de l'efficacité de la documentation.
- La possibilité de créer automatiquement un texte de résumé lisible.
- Utile pour les assistants personnels tels qu'Alexa.
- Permet à une organisation d'utiliser des chatbots pour le support client.
- Analyse des sentiments plus facile.

3.4. Défis associés à la NLP

La NLP n'est pas encore totalement perfectionnée. Par exemple, l'analyse sémantique peut encore être un défi pour la NLP. D'autres difficultés incluent le fait que l'utilisation abstraite du langage est généralement difficile à comprendre pour les programmes. Par exemple, la NLP ne capte pas facilement le sarcasme. Ces sujets nécessitent généralement la compréhension des mots utilisés et du contexte dans lequel ils sont utilisés. Autre exemple, une phrase peut changer de sens en fonction du mot sur lequel le locuteur met l'accent. La NLP est également mise au défi par le fait que la langue et la manière dont les gens l'utilisent changent continuellement.[42]

4. Conclusion:

Dans ce chapitre, nous avons appris et présenté les principaux concepts de l'intelligence artificielle et du traitement de langage naturel donc après avoir connu les concepts de base de ces deux nous pouvons commencer à construire notre modèle.

CHAPITRE3 : CONTRIBUTION ET RESULTATS

1. Introduction

Dans ce chapitre, nous exposons la phase de réalisation de notre projet. Cette phase est considérée comme étant la concrétisation finale de notre projet représentée par le synthèse et génération in silico des molécules biochimiques en basant sur les méthodes d'intelligence computationnelle. Tout d'abord, nous allons expliquer la démarche et la méthode proposée, ensuite nous détaillons l'implémentation de l'approche proposée ainsi que les outils utilisés pour illustrer le fonctionnement du modèle. Enfin, nous discutons les résultats obtenus par notre modèle et les comparer avec d'autres approches dans la littérature.

2. Approche proposée

Dans cette section, nous allons expliquer l'approche proposée pour la génération de molécule autrement dit inverse-QSAR en utilisant le Deep Learning et le traitement de langage naturelle. Notre approche se compose de trois étapes, dont la première est le traitement de données SMILES en utilisant le modèle de Deep Learning embarqué. La deuxième étape consiste à entrainer un modèle de réseaux de neurones convolutif en utilisant des bases de données QSAR afin de valider la génération de molécules. La troisième étape consiste à utiliser le modèle Deep Learning proposé afin de générer *de novo* molécules ont l'activité biologique dédiée. La figure 3.1 illustre l'approche proposée.

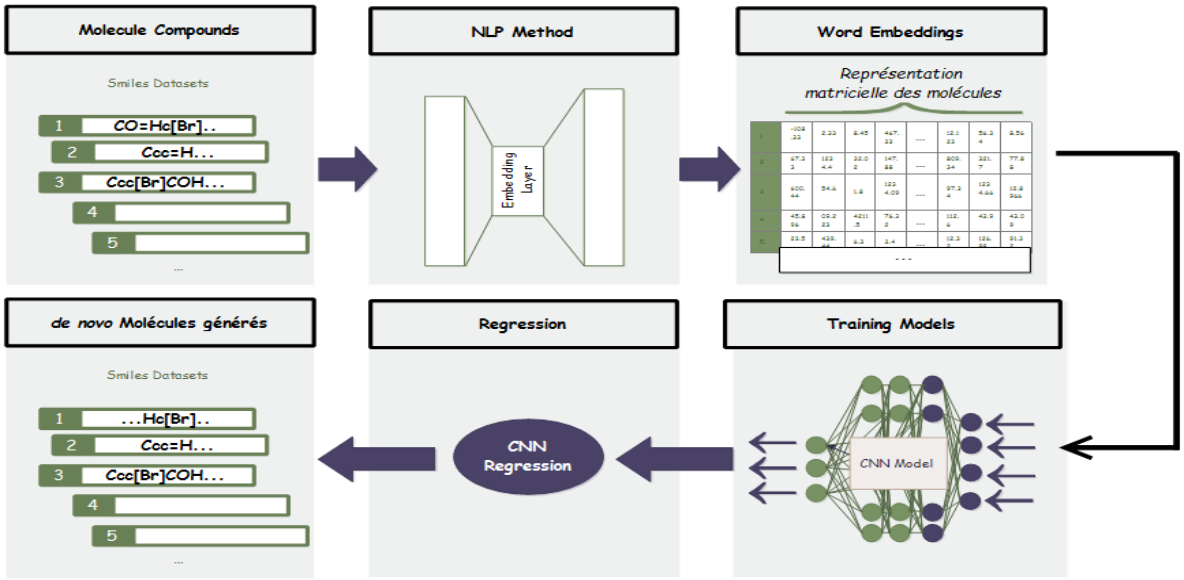


Figure 3. 1 Illustration sur l'approche proposée

2.1. Traitement de données SMILES

Dans cette étape, nous allons transformer les données SMILES en données numériques qui représente la sémantique de chaque molécule. Pour cela, nous avons utilisé un traitement de langage naturelle (NLP). Dans cette étape, nous suivons les étapes suivantes :

- **Génération d'un vocabulaire :** Dans cette étape, nous allons construire un vocabulaire contenant tous les sous-structures des molécules pour un jeu de données. Le vocabulaire sera le corpus de la prochaine étape qui est défini comme suit :

$$Vocabulaire\ (SMILES_{dataset}) = C_1, C_2, C_3, \dots, C_n \quad (1)$$

Où, *C* et *n* sont les sous-structures et le nombre de possibilités de concaténation d'atomes respectivement. L'ensemble de données de grande complexité, qui contient environ 700000 composés tirés du tableau de bord CompTox de l'EPA (Agence de protection de l'environnement) (https://comptox.epa.gov/dashboard/), a été utilisé pour construire le vocabulaire, les composés de diversité permettent de construire un très grand corpus qui comprend toutes les combinaisons de mots des trois ensembles de données. Ce vocabulaire généré évaluera le modèle Word2Vec afin de prédire la représentation vectorielle optimale des composés chimiques. En effet, le modèle généré fera la distinction entre les représentations

d'enrobage de molécules composés avec une grande précision. Ci-dessus un tableau contenant un exemple d'extraction des sous-structures.

Table 3. 1 Extraction de sous-structures d'un jeu de données SMILES

Index	Molécules	Composants	Taille
0	O=Cc1ccco1	[O, O=C, O=Cc1ccco1, C, Cc1ccco1, c1ccco1]	6
1	Cc1ccc(O)cc1	[C, Cc1ccc(O)cc1, c1ccc(O)cc1, O]	4
2	OCc1ccccn1	[N, NC, NCc1ccccc1, C, Cc1ccccc1, c1ccccc1]	6

- **Apprentissage d'un modèle Word2Vec :**

Nous avons formé un modèle Word2Vec pour transformer les molécules SMILES en représentation vectorielle. Cet algorithme d'intégration de mots est l'un des meilleurs algorithmes de traitement du langage naturel à ce jour. Le modèle Word2vec est un réseau de neurones à deux couches qui vise à traiter des mots pour prédire les représentations vectorielles. Dans ce sujet, les mots qui partagent des contextes similaires sont représentés par des vecteurs numériques proches. Nous avons utilisé le vocabulaire généré à l'étape précédente pour entraîner le modèle embarqué afin d'obtenir un transformateur entraîné noté *Vec(x)*, où *X* est le corpus généré. Ce transformateur mappe un mot *x* de vocabulaire dans un espace vectoriel continu de taille *d*.

$$Vec : Vocabulaire \rightarrow \mathbb{R}^d \quad (2)$$

$$X \rightarrow Vec\,(x) \quad (3)$$

- **Transformation de données SMILES :**

Le modèle embarqué proposé formé, nous prédisons le vecteur de représentation numérique de SMILES. Tout d'abord, afin de pouvoir prédire la vectorisation de la molécule SMILES, cette dernière est découpée en sous-structures de composés. Ensuite, nous utilisons l'équation (3) pour calculer le vecteur de représentation numérique de l'ensemble SMILES comme suit :

$$Vec(SMILE_{molecule}) = \sum Vec\,(X1);\ Vec\,(X2);\ \ldots;\ Vec(Xi))$$

Où, molécule *SMILES = concaténation (X1, X2 .... Xi)* et *i* est le nombre de concaténations de longueur L dans la molécule SMILES. L'ensemble du prétraitement des données est illustré à la figure 3.2.

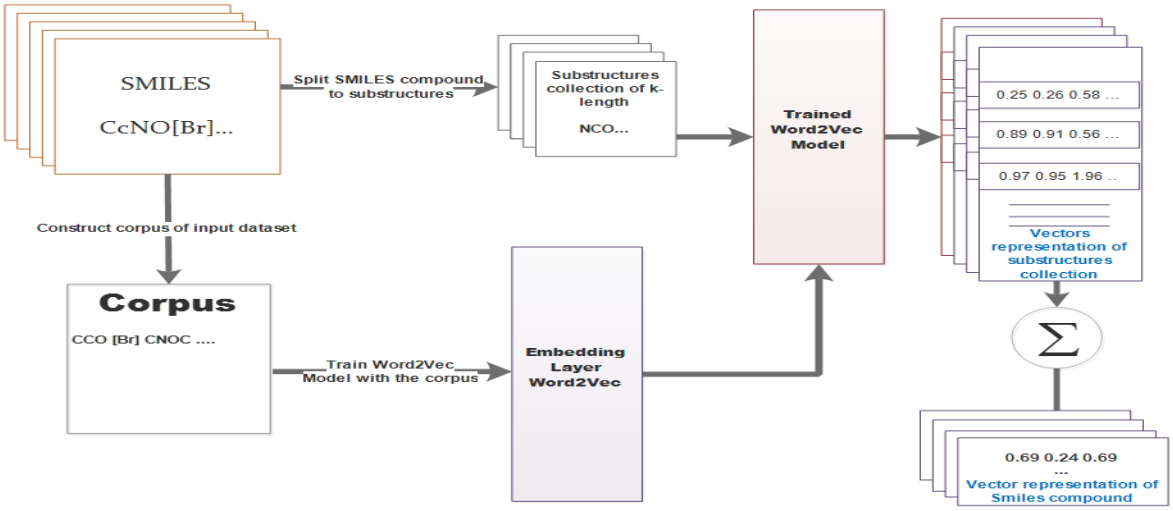


Figure 3. 2 Processus de prétraitement de données

2.2. Modèle d'apprentissage

Les données recueillies sur des périodes successives telles que les composés moléculaires sont caractérisés comme une série chronologique. Dans ce cas, LSTM est une approche intéressante pour traiter ce type de données. Dans ce type d'architecture d'apprentissage en profondeur, le modèle passe l'état caché précédent à l'étape suivante de la séquence, dans laquelle l'ordre des données est extrêmement important.

De l'autre côté, les réseaux de neurones convolutifs sont les meilleurs modèles d'apprentissage en profondeur pour extraire le modèle de caractéristiques à partir de données représentées sous forme de matrices telles que des images. Dans ce sujet, la couche convolutive vise à extraire la carte des caractéristiques de la représentation vectorielle des composés moléculaires.

Pour prédire la toxicité ou la propriété d'un composé moléculaire, un modèle de mémoire convolutive à long terme (ConvLSTM) a été utilisé. Ce dernier est une extension du RNN de mémoire à long terme (LSTM) populaire. Dans ce modèle, les nœuds entièrement connectés du module LSTM sont remplacés par des portes convolutives, ce qui le rend capable de coder des caractéristiques spatio-temporelles de la représentation vectorielle SMILES. L'architecture proposée du modèle est illustrée dans la figure 3.3.

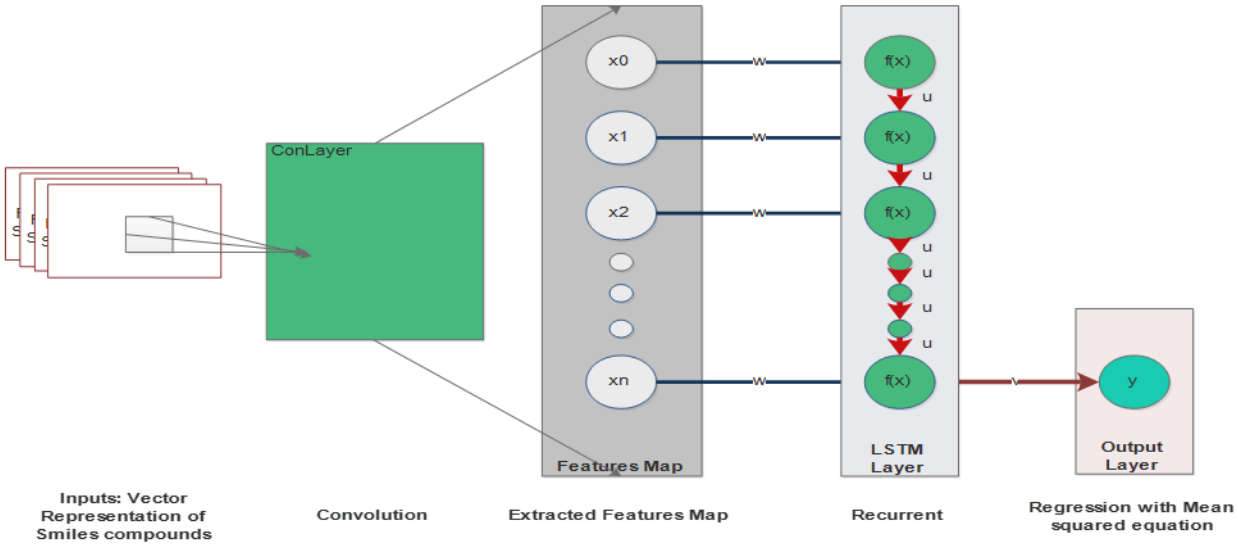


Figure 3. 3 L'architecture de modèle ConvLSTM

1- **Étapes de convolution :** L'extraction du modèle de caractéristiques à partir de la représentation de vecteurs numériques prédite des fragments SMILES est effectuée à l'aide de couche convolutive, qui contiennent un ensemble de filtres dont les paramètres doivent être appris à partir des vecteurs d'entrée pour obtenir une carte de caractéristiques.

2- **Rectification :** La fonction des unités linéaires rectifiées (*ReLu*) a été utilisée dans les vecteurs d'entrée en raison de son insaturation et du gradient élevé si les nœuds des couches sont activés. La fonction ReLu est définie comme suit :

$$ReLu\ (x)\ =\ max\ (0, x)\ (5)$$

3- **LSTM :** La couche LSTM a une structure en chaîne, avec une structure différente de module répétitif. Au lieu d'avoir une seule couche de réseau neuronal, il existe de nombreuses interactions d'une manière très spéciale. Après avoir extrait la carte des caractéristiques de la représentation vectorielle des composés moléculaires à l'aide de la couche convolutionnelle, nous avons intégré une couche LSTM pour traiter cette carte des caractéristiques sous forme de chaîne. Après cette étape, LSTM peut ajouter des informations ou supprimer toute information inutile pour prédire l'activité de ces composés.

4- **Dropout** : Les couches d'abandon remettent à zéro de manière aléatoire les entrées de la couche de sommets suivante pendant l'apprentissage avec une probabilité choisie de 0,5. Cela régularise le réseau et évite le surajustement (overfitting).

5- **Fonction de perte :** Nous avons utilisé la fonction de l'erreur quadratique moyenne MSE pour mesurer la divergence entre les distributions de probabilités correspondant à l'attribution du fragment SMILES à une valeur de toxicité. La fonction de perte comme l'équation suivante :

$$MSE = \frac{1}{N} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \quad (6)$$

Où, y_i est l'observé expérimentalement et $\hat{y}_i$ est le prédit théoriquement, je suppose que n est le nombre de nœuds de sortie.

6- **Algorithme d'optimisation d'apprentissage :** Pour améliorer l'apprentissage du modèle proposé, nous avons intégré l'algorithme d'optimisation ADAM (Adaptive Moment Estimation) grâce à ses performances de calcul. Il vise principalement à ajuster les paramètres d'apprentissage lors de l'entraînement du réseau de neurones (figure 3.4). Tous les modèles de processus de formation sont donnés dans le tableau 3.2.

Table 3. 2 Paramétrage de modèle ConvLSTM proposé

Layer (type)	**Output Shape**	**Param #**
conv_lst_m2d_2 (ConvLSTM2D)	(None, 31, 7, 128)	264704
max_pooling2d_2 (MaxPooling2)	(None, 15, 3, 128)	0
dropout_2 (Dropout)	(None, 15, 3, 128)	0
flatten_2 (Flatten)	(None, 5760)	0

dense_1 (Dense)	(None, 16)	92176
dropout_3 (Dropout)	(None, 16)	0
dense_2 (Dense)	(None, 1)	17
Total params: 356,897 Trainable params: 356,897 Non-trainable params: 0		

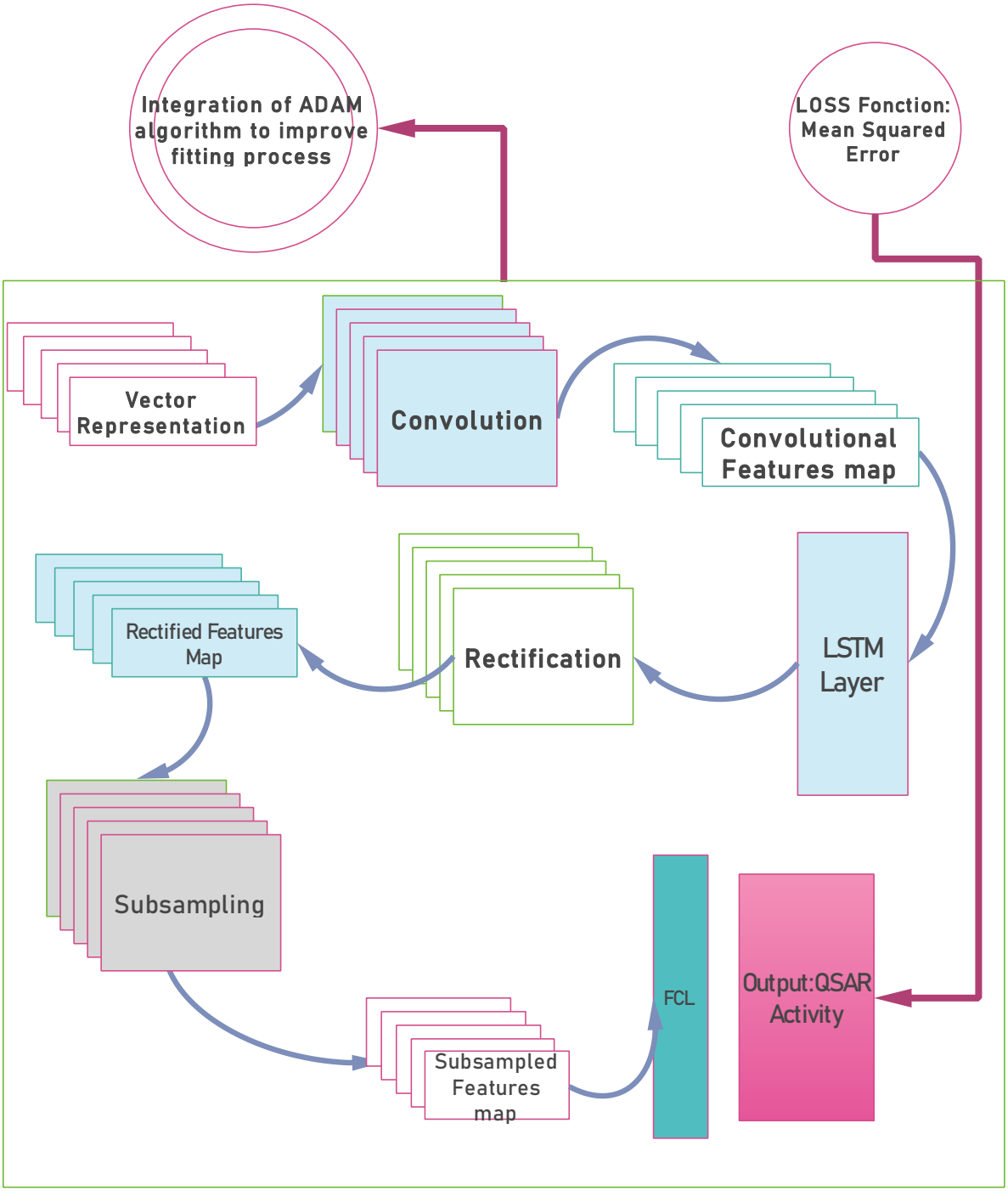


Figure 3. 4 Le processus d'apprentissage d'activité biologique

2.3. Génération de données

Dans la dernière étape, nous voulons générer de novo molécules avec l'activité adéquate, pour cela nous suivant une technique d'optimisation stochastique. Ce processus suit les étapes suivantes :

1- Charger le vocabulaire déjà construit dans l'étape de prétraitement de données.

2- Choisir des sous-structures aléatoirement à partir le vocabulaire généré afin de synthétiser une nouvelle molécule. Cette étape est formalisée par l'équation suivante :

$$Molécule_n = Random(Vocabulaire(X))$$

3- Prédire la représentation numérique de la molécule générée.

4- La fonction objective de méthode proposé est comme suit :

$$Fit(Molécule_n) = \left(Vec(Molécule_n) - Vec(Molécule_0)\right)^2$$

Ou Molécule0 est une molécule avec l'activité biologique existe déjà dans la base de données d'apprentissage.

5- Si la fitness *Fit(Molécule$_n$)* égale ou approche à 0, prédire l'activité biologique de *Molécule$_n$* En utilisant le modèle ConvLSTM déjà entrainé. Sinon choisir une autre sous-structure aléatoirement à la place d'une autre.

3. Jeux de données

Pour évaluer la performance de notre approche, les deux ensembles de données de toxicité suivants ont été utilisés dans ce travail :

- **Données sur la DL50 chez le rat :** la toxicité orale aiguë, qui est exprimée en DL50 de la dose létale médiane, est l'un des paramètres toxicologiques les plus importants à évaluer dans la découverte de médicaments. La DL50 est la dose d'un produit chimique qui tue la moitié des rats traités, les valeurs ont été exprimées en (pLD50 mol / kg). L'ensemble de données utilisé dans cette étude contient 7314 composés rapportés par plusieurs travaux.
- **Base de données Tetrahymena pyriformis IGC50** : ce point final est l'un des ensembles de données les plus couramment utilisés dans la modélisation QSAR, pour évaluer la toxicité aqueuse des composés. l'IGC50 est la plus grande quantité d'informations sur la toxicité aqueuse, qui est testée dans un seul laboratoire par une méthode unique, fiable et robuste. La toxicité est exprimée par la concentration inhibitrice de croissance à 50% (pIGC50 mol / L) de l'organisme T. pyriformis après 40 heures. Les données contenant 1792 composés ont été obtenues à partir du logiciel QSAR Toolbox (http://oasis-lmc.org).

Table 3. 3 Base de données utilisé afin de valider l'approche proposée

Dataset	Propriété	Taille
T. pyriformis IGC_{50}	Aqueous toxicities	1792
Rat LD_{50}	Acute oral toxicity	7413

4. Métriques d'évaluation

Afin d'évaluer et de comparer les performances de l'approche proposé, l'évaluation des performances prédictives de notre modèle de régression repose principalement sur trois statistiques, à savoir l'erreur absolue moyenne (MAE), l'erreur quadratique moyenne (RMSE) et le carré coefficient de corrélation (R2) pour les ensembles internes et externes. Ils ont été définis comme :

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{8}$$

$$R^2 = 1 - \frac{\frac{1}{N}\sum_{i=1}^{ntr}(Y_i - \hat{Y}_i)^2}{\frac{1}{N}\sum_{i=1}^{ntr}(Y_i - \bar{Y}_{tr})^2} \tag{9}$$

$$R^2_{ext} = 1 - \frac{\frac{1}{N}\sum_{i=1}^{next}(Y_i - \hat{Y}_i)^2}{\frac{1}{N}\sum_{i=1}^{next}(Y_i - \bar{Y}_{tr})^2} \tag{10}$$

Où y_i est l'observé expérimentalement et $\hat{y}_i$ est le théoriquement prédit entre la molécule i et la molécule correspondante, $\bar{Y}\_i$ était la moyenne de la valeur expérimentale, n est le nombre de points de données tr pour l'ensemble d'apprentissage et ext pour l'ensemble externe.

5. Paramètres de réglage

Dans l'étape de traitement des données, nous avons intégré l'algorithme word2vec pour prédire la représentation embarquée de mots du composé SMILES. Nous avons construit un corpus qui comprend toutes les sous-structures à partir d'un ensemble de données comprenant plus de 700000 composés. La taille du vecteur prédite a été fixée à 128 et remodelée en un vecteur à 2 dimensions de (16,8) pour le modèle CNN.

Dans l'étape du modèle d'apprentissage, le modèle d'apprentissage est un réseau de neurones à convolution, où nous fixons arbitrairement le nombre de noyaux à 32 et la taille de ces noyaux à (2,1) pour la couche de convolution et à (1,2) pour la couche de sous-échantillonnage.

6. Plateformes logicielles & Hardware utilisés pour l'implémentation :

6.1. Hardware :

Pour faire un apprentissage d'un réseau Deep et pouvoir construire un modèle, il faut disposer de moyens aptes a faire cette tache tout en étant capable de fournir les ressources nécessaires pour le bon déroulement de ce processus. Ces moyens sont répartis entre hard et soft qu'on va décrire ci-après :

Type : station de Travail

Nom : HP Z440

Mémoire Vive (RAM) : 32GO

Processeur : Intel® Xeon® Processor E5-1620 v3 (3.5GHz , 4coers , Intel® vPro™)

Graphique 1 : Intel (R) HD Graphics Family

Carte graphique 2 (GPU) : NVIDIA GTX 780 (3 Go GDDR5)

Disque Dure HDD : 1TO

6.2. SOFTWARE:

Au cours de cette étape, nous soulignerons les divers outils et plateformes que nous Avons utilisés afin de mener à bien notre projet concerné l'utilisation du Deep Learning pour la génération des molécules :

Python est un langage de programmation open source interprété côté serveur et non compilé. Créé par Guido van Rossum [39], il est utilisé pour le développement web, le développement de jeux vidéo et autres logiciels, ainsi que pour les interfaces utilisateur graphique. Il a notamment été utilisé dans la création d'Instagram, de Youtube et de Spotify, et est l'un des langages de programmation officiels de Google.

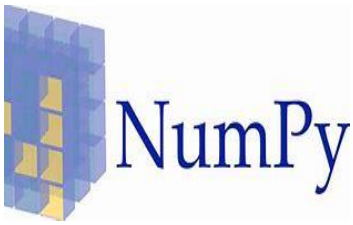
Google Colaboratory, parfois appelé Colaboratory en abrégé, est un service cloud de Google qui réplique Jupyter Notebook dans le cloud. Son utilisation ne demande aucune installation. IL est principalement destiné pour créer et partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif.

Jupyter Notebook est une application Web open source permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. Les utilisations comprennent: le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique et bien plus encore.

TensorFlow est une bibliothèque de logiciels open source pour le calcul numérique à l'aide de graphiques de flux de données. Il a été initialement développé par l'équipe Google Brain au sein de l'organisation de recherche Machine Intelligence de Google pour l'apprentissage automatique et la recherche sur les réseaux de neurones profonds, mais le système est suffisamment général pour être applicable dans une grande variété d'autres domaines également.

Keras est une bibliothèque Python open-source de réseaux neuronaux fonctionner par défaut avec TensorFlow (mais capable de fonctionner aussi sur, Theano, PlaidML et autres), fournit à l'adress : https://keras.io/. Elle peut être fonctionner sur GPU ou CPU

Pandas est un outil de manipulation de données de haut niveau développé par Wes McKinney. sa structure de données clés sont des Series ou des DataFrame. Les DataFrames noua permettent de stocker et de manipuler des données tabulaires dans une vue bidimensionnelle.



NumPy (est une bibliothèque python open source pour les calculs numériques créé en 2005 par Travis Oliphant utilisée pour travailler avec les tableaux et les fonctions dans le domaine de l'algèbre linéaire, la transformation des Fourier et des matrices .

Gensim est une bibliothèque Python pour la modélisation de sujets, l'indexation de documents et la recherche de similitudes avec de grands corpus. Le public cible est la communauté du traitement du langage naturel (NLP) et de la recherche d'informations (IR).

RDKit est une boîte à outils de chimio-informatique open-source écrite en C ++ qui est également utilisable depuis Java ou Python. Il comprend une collection de fonctionnalités standard de chimio-informatique pour les E / S moléculaires, la recherche de sous-structures, les réactions chimiques, la génération de coordonnées (2D ou 3D), la prise d'empreintes digitales, etc.

PubChemPy fournit un moyen d'interagir avec PubChem en Python. Il permet des recherches chimiques par nom, sous-structure et similitude, la standardisation chimique, la conversion entre les formats de fichiers chimiques, la représentation et la récupération des propriétés chimiques.

Scikit-learn est une bibliothèque libre Python destinée `a l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria et Telecom ParisTech.

Microsoft Office est une suite d'applications de productivité de bureau spécialement conçue par Microsoft pour une utilisation professionnelle. Il s'agit d'un produit propriétaire de Microsoft Corporation et a été lancé pour la première fois en 1990. Pendant des décennies, MS Office a été un modèle dominant dans la fourniture d'environnements logiciels modernes de gestion de documents de bureau.

7. Résultats expérimentaux

Avec la base de données IGC50 de 1792 composés et les données Rat LD50 de 7314 composés, les performances prédictives de trois modèles d'apprentissage automatique ont été

évaluées en utilisant 80% des données comme ensemble d'apprentissage et 20% des données comme ensemble de test. Les vecteurs composés dérivés résultant d'un modèle d'inclusion formé ont été utilisés comme caractéristiques afin de prédire les valeurs de toxicité dans les trois modèles d'apprentissage automatique.

L'incorporation de mots en profondeur est l'une des méthodes les plus utilisées dans le domaine du traitement du langage naturel, en raison de son efficacité à prédire les vecteurs numériques qui représentent les caractéristiques des mots. Dans notre approche, nous avons appliqué un traitement d'incorporation de mots à l'aide de Word2vec pour prédire les fonctionnalités SMILES à l'aide d'un corpus généré contenant toutes sous-structures à partir d'un ensemble de données comprenant plus de 700000 composés. L'entraînement de notre modèle word2vec avec ce corpus a pu générer une fonction de prédiction dans un espace continu en raison du grand nombre de mots dans le corpus. Tout en utilisant une petite taille de corpus, le modèle word2vec généré prédit une représentation numérique de mots dans un espace discret qu'il influencera sur les performances des régresseurs d'apprentissage automatique. En effet, les résultats obtenus ont montré qu'en utilisant notre approche, nous avons obtenu de meilleures performances. Aussi, nous avons pu générer de novo molécules avec des valeurs d'activité prometteuses.

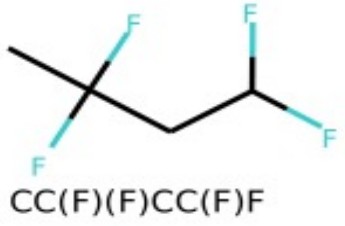
Les tableaux suivants présentent les résultats expérimentaux obtenus.

Table 3. 4 Tableau des résultats expérimentaux

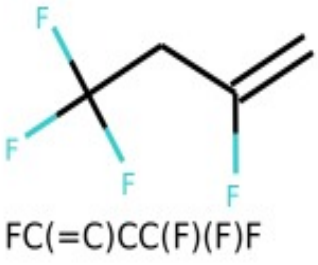
Machine learning technique	**Training set**	**Test set**		
	R^2	**R^2**	**RMSE**	**MAE**
IGC50	0.99	0.89	0.29	0.18
LD50	0.97	0.85	0.32	0.26

Ainsi nous avons pu générer six molécules pour les deux bases de données utilisées :-dessus :

Résultats de génération (IGC50):



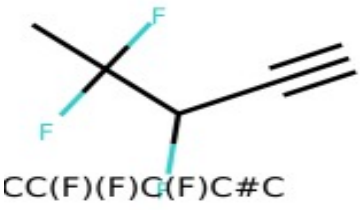
4.01



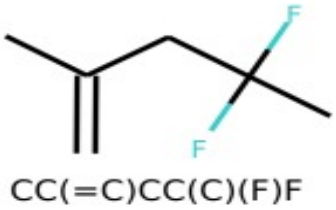
5.38

3.98

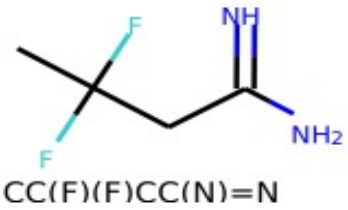
Résultats de génération (LD 50):



6.87



13.18



2.30

8. Conclusion

Dans ce chapitre, nous avons présenté notre contribution dont nous avons décrit la méthode proposée, l'implémentation, les différents outils utilisée et les différentes bibliothèques. Finalement, nous Avons présenté les résultats obtenus qui sont très satisfaisants.

CONCLUSION GENERALE

Conclusion générale :

Dans notre travail de fin d'étude, nous avons abordé le problème de la génération des molécules bioactives. Nous avons mis l'accent sur l'importance de la découverte des substances biologiquement actives dans le traitement du cancer.

Dans un premier temps nous avons décrit les principaux concepts de la chimio-informatique et les techniques d'intelligences computationnelles utilisées.

Nous avons proposé une nouvelle approche, dite inverse QSAR basé sur le Deep Learning et le traitement naturel de texte. Cette approche a suivi trois étapes :

- la première étape est le traitement des jeux de données SMILES en utilisant le traitement naturel de langage avec le Deep Learning.
- la deuxième étape, nous avons entraîné un modèle de réseaux de neurones convolutifs en utilisant les vecteurs numériques générés de l'étape précédente où nous avons obtenu des résultats de performances très prometteuse.
- La troisième étape nous avons proposé une méthode de recherche stochastique et itérative afin de générer de novo molécule avec l'activité ciblée.

Les résultats de performance ont montré que notre approche atteint un coefficient de détermination égale à 89% pour les données IGC50 et 85% pour les données de LD50.

REFERENCES

Références

[1] Reddy, A. S., Chen, L. & Zhang, S. in De novo Molecular Design (ed. Schneider, G.). 97–124 (Wiley, Hoboken, 2013).

[2] Durrant, J. D. & Amaro, R. E. in De novo Molecular Design (ed. Schneider, G.) 125–142 (Wiley, Hoboken, 2013).

[3] Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of druglike molecules based on molecular complexity and fragment contributions.
J. Cheminform. 1, 1–11 (2009).

[4] N. P. Vishwakarma, Cheminformatics, Journal of Proteomics & Bioinformatics(January 2010)

[5] A. Höskuldsson, J. Chemometrics, 2, 211 (1988).

[6] J. A. Burns, G. M. Whiteside. Chem. Rev., 93, 2583 (1993).

[7] S.E. Manahan, Toxicological chemistry and biochemistry, 3rd ed. CRC Press (2003).

[8] V. Y. Nalimov. The Application of Mathematical Statistics to Chemical Analysis, Addison- Wesley, Reading, MA (1962).

[9] R. Calcutt, R. Body. Statistics for Analytical Chemists. Champman & Hall, New York (1983).

[10] J. C. Miller, J. N. Miller. Statistics for Analytical Chemistry. Ellis Horwood, New York (1988).

[11] P. C. Meier, R. E. Zund. Statistical Methods in Analytical Chemistry. Wiley, New York (1993).

[12] P. Dagnélie. Statistique Théorique Et Appliquée. Tomes 1 et 2. De Boeck & Larcier s. a. (1998).

[13]- R. Tomassone, E. Lesquoy, C. Miller. La régression : nouveaux regards sur une ancienne méthode statistique. Masson, INRA (1983).

[14] R. Wehrens, H. Putter, L. M. C. Buydens. Chemom. Int. Lab. Syst., 54, 35- 52 (2000).

[15] L. Eriksson, J. Jaworska, A. P. Worth, M. T. D. Cronin, R. M. Mc Dowell, P. Gramatica. Methods for RelAIbility and Uncertainty Assessment and for Applicability

Evaluations of Classification and Regression Based QSARs. Environmental Health Perspectives 111, 1361-1375 (2003).

[16](en ligne). Consulté le 23 juillet 2020. «https://www.lebigdata.fr/data-mining-definition-exemples»

[17] Kappel JC, Fan YC, Lam KS. Application of the "libraries from libraries" concept to "one-bead one-compound" combinatorAIl chemistry. Adv Exp Med Biol.

[18] Moos WH, Hurt CR, Morales GA. CombinatorAIl chemistry: oh what a decade or two can do. Mol Divers..

[19] Weber L. High-diversity combinatorAIl libraries. Curr Opin Chem Biol.

[20] Bearden AP, Schultz TW. 1997. Environ. Toxicol. Chemistry.

[21]Schultz TW, Lin DT, Arnold LM. 1991. The Science of the Total Environment.

[22] « 2.5.1 Two-dimensional (2-D) similarity methods | DivCHED CCCE: Cheminformatics OLCC ». Consulté le 24 juillet 2020. http://olcc.ccce.divched.org/2015OLCCModule6P1TLO-2-5-1.

[23] Bender, Andreas, et Robert C. Glen. « Molecular Similarity: A Key Technique in Molecular Informatics ». Organic & Biomolecular Chemistry 2, no 22 (21 novembre 2004): 3204-18. https://doi.org/10.1039/B409813G.

[24] Hansen OC. Quantitative structure-activity relationships (QSAR) and pesticides: Ministry of the Environment, Environmental Protection Agency; 2004.

[25] Brown AC, Fraser TR. V.—On the connection between chemical constitution and physiological action. Part. I.—On the physiological action of the salts of the ammonium bases,derived from strychnAI, brucAI, thebaAI, codeAI, morphAI, and nicotAI. Earth and Environmental Science Transactions of The Royal Society of Edinburgh. 1868;25(1):151-203.

[26] Free SM, Wilson JW. A mathematical contribution to structure-activity studies. Journal of Medicinal Chemistry. 1964;7(4):395-9.

[27] Xiong, J. (2006). EssentAIl bioinformatics.

[28] C. D. Selassie, "History of quantitative Structure-Activity relationships".

[29] H. Waterbeemd, S. Rose, "Quantitative approaches to structure-activity relationships", in *Book* "Quantitative approaches to structure-activity relationships". 2003

[30] Gyanendra Singh Ajitanshu Mishr,Dheeraj Sagar , AN OVERVIEW OF ARTIFICAIL INTELLIGENCE

[31] En Ligne. Consulté le 12 aout 2020
https://www.techslang.com/what-are-the-basic-concepts-in-ai/
[32] En Ligne. Consulté le 12 aout 2020.
https://en.proft.me/2015/12/24/types-machine-learning-algorithms/
[33] Deng, L., & Yu, D. (2014). Deep Learning : Methods and Application.
[34] En Ligne. Consulté le 16 aout 2020.
https://www.mygreatlearning.com/blog/deep-learning-applications/
[35]En Ligne. Consulté le 16 aout 2020.
https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks/
[36] En Ligne. Consulté le 16 aout 2020.
https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network
[37]En Ligne. Consulté le 24 aout 2020.
https://blog.paperspace.com/autoencoder-image-compression-keras/
[38] En Ligne. Consulté le 16 aout 2020.
https://deepai.org/machine-learning-glossary-and-terms/autoencoder
[39]En Ligne. Consulté le 24 aout 2020
https://deepai.org/machine-learning-glossary-and-terms/recurrent-neural-network
[40]En Ligne. Consulté le 24 aout 2020
https://searchbusinessanalytics.techtarget.com/definition/natural-language-processing-NLP
[41]En Ligne. Consulté le 24 aout 2020
https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec/
[42] En Ligne : Consulté le 25 aout 2020.
https://www.investopedAI.com/terms/a/artificAIl-intelligence-ai.asp

RESUME

Résumé

Le défi de trouver des substances bioactives pour traiter, guérir, prévenir ou diagnostiquer une maladie est de trouver des molécules qui répondent à de nombreuses limitations du profil métabolique souhaité.

L'examen expérimental de milliers de particules malgré l'utilisation d'équipements robotiques reste coûteux et prend du temps, ainsi que les méthodes d'examen par défaut et les nouvelles méthodes de conception de particules basées sur la structure, bien qu'elles soient plus rapides, restent difficiles lorsque nous n'avons aucune information préalable.

Les capacités fournies par l'intelligence artificielle aux machines pour la résolution de problèmes et la prise de décision intelligentes ont piqué notre intérêt.

Dans notre travail de mémoire, nous proposons une nouvelle approche inverse QSAR basé sur le Deep Learning et le traitement naturelle de texte. Notre approche suivre trois étapes dont, la première est le traitement des jeux de données SMILES.

Dans cette étape, nous générons un vocabulaire contenant tous les sous-structures de jeux de données d'entrée. Ensuite, nous allons entraîner un modèle de Deep Learning embarqué afin de prédire et transformer les données en données vectorielles numériques.

Dans la deuxième étape, nous avons entraîné un modèle de réseaux de neurones convolutifs en utilisant les vecteurs numériques générés de l'étape précédente. Cette étape, va générer une fonction de régression pour prédire les activités biologiques. Dans la troisième étape, nous proposons une méthode de recherche stochastique et itérative afin de générer de novo molécule avec l'activité ciblé. Les résultats de performance montrent que notre approche atteint un coefficient de détermination égale à 89% pour les données IGC50 et 85% pour les données de LD50.

ملخص

يتمثل تحدي العثور على مواد نشطة بيولوجيًا لعلاج مرض ما، الوقاية منه أو تشخيصه في العثور على الجزيئات التي تلبي الخصائص المطلوبة.

لا يزال الفحص التجريبي لآلاف الجزيئات بالرغم من استخدام المعدات الروبوتية باهظ التكلفة ويستغرق وقتًا طويلاً ، حتي بالنسبة لطرق الفحص الافتراضية وطرق تصميم الجسيمات الحديثة القائمة على الهياكل الكيميائية ، على الرغم من أنها أسرع ، الا انها تظل صعبة عندما لا تكون لدينا معلومات مسبقة.

لقد أثارت القدرات التي يوفرها الذكاء الاصطناعي للآلات لحل المشكلات واتخاذ القرارات بذكاء اهتمامنا.

في عمل أطروحتنا، نقترح منهج QSAR معكوس جديد يعتمد على التعلم العميق والمعالجة الطبيعية للكلمات. يتبع نهجنا ثلاث خطوات، أولها معالجة مجموعات بيانات.SMILES

في هذه الخطوة، نقوم بإنشاء مفردات تحتوي على جميع التركيبات الفرعية لمجموعات بيانات الإدخال. بعد ذلك، سنقوم بتدريب نموذج التعلم العميق المضمن للتنبؤ بالبيانات وتحويلها إلى بيانات متجه رقمية.

في الخطوة الثانية، نقوم بتدريب نموذج للشبكات العصبية التلافيفية باستخدام المتجهات الرقمية التي تم إنشاؤها من الخطوة السابقة. ستولد هذه الخطوة دالة انحدار للتنبؤ بالأنشطة البيولوجية. في الخطوة الثالثة، نقترح طريقة بحث عشوائية وتكرارية من أجل توليد جزيء de novo مع النشاط المستهدف. تظهر نتائج الأداء أن نهجنا يحقق معامل تحديد يساوي 89٪ لبيانات IGC50 و85٪ لبيانات.LD50 .

Abstract

The challenge of finding bioactive substances to treat, cure, prevent, or diagnose disease is to find molecules that meet many limitations of the desired metabolic profile.

Experimental examination of thousands of particles despite the use of robotic equipment remains expensive and time-consuming, as do default examination methods and newer structure-based particle design methods, although they are faster, remain difficult when we have no prior information. The capabilities artificial intelligence provides to machines for intelligent problem solving and decision making have piqued our interest.

In our thesis work, we propose a new inverse QSAR approach based on Deep Learning and natural word processing. Our approach follows three steps, the first of which is the processing of SMILES datasets.

In this step, we generate a vocabulary containing all the substructures of the input datasets. Next, we will train an embedded Deep Learning model to predict and transform the data into numerical vector data.

In the second step, we have trained a convolutional neural network model using the numerical vectors generated from the previous step. This step will generate a regression function to predict the biological activities. In the third step, we propose a stochastic and iterative research method in order to generate de novo molecule with the targeted activity. The performance results show that our approach achieves a coefficient of determination equal to 89% for the IGC50 data and 85% for the LD50 data.

Présenté par :
BADAOUI Mohamed Yahia Zeriab

Génération *in-silico* des molécules visées thérapeutiques basée sur les méthodes d'intelligence computationnelle

Mémoire de fin de cycle en vue de l'obtention du diplôme de Master en Biochimie Appliquée.

Résumé :

Le défi de trouver des substances bioactives pour traiter, guérir, prévenir ou diagnostiquer une maladie est de trouver des molécules qui répondent à de nombreuses limitations du profil métabolique souhaité.

L'examen expérimental de milliers de particules malgré l'utilisation d'équipements robotiques reste coûteux et prend du temps, ainsi que les méthodes d'examen par défaut et les nouvelles méthodes de conception de particules basées sur la structure, bien qu'elles soient plus rapides, restent difficiles lorsque nous n'avons aucune information préalable.

Les capacités fournies par l'intelligence artificielle aux machines pour la résolution de problèmes et la prise de décision intelligentes ont piqué notre intérêt.

Dans notre travail de mémoire, nous proposons une nouvelle approche inverse QSAR basé sur le Deep Learning et le traitement naturel de texte. Notre approche suivre trois étapes dont, la première est le traitement des jeux de données SMILES.

Dans cette étape, nous générons un vocabulaire contenant tous les sous-structures de jeux de données d'entrée. Ensuite, nous allons entraîner un modèle de Deep Learning embarqué afin de prédire et transformer les données en données vectorielles numériques.

Dans la deuxième étape, nous avons entraîné un modèle de réseaux de neurones convolutifs en utilisant les vecteurs numériques générés de l'étape précédente. Cette étape, va générer une fonction de régression pour prédire les activités biologiques. Dans la troisième étape,
nous proposons une méthode de recherche stochastique et itérative afin de générer de novo molécule avec l'activité ciblé. Les résultats de performance montrent que notre approche atteint un coefficient de détermination égale à 89% pour les données IGC50 et 85% pour les données de LD50.

Laboratoires de recherche :

- Laboratoire de de biochimie appliqué, Département de Biochimie et de Biologie Moléculaire et Cellulaire, U. Frères Mentouri – Constantine 1
- Unité bio-informatique et bio-statistique (BIBS-U), centre de recherche en biotechnologie - Constantine

Jury d'évaluation :

Président du jury : Mr. BENSEGUENI ABDERRAHMANE	Pr.	UFM, Constantine 1
Encadreur: Mr. DEMS MOHAMED ABDESSELEM.	MRA	CRBt Constantine
Co- Encadreur : Mr. BOUKLIA ABDELBASSET	DR	CRBt Constantine
Examinateur : Mr. MOKRANI ELHASSEN	MAA	UFM, Constantine 1

Date de soutenance : 05/10/2020